

ภาคผนวก

ภาคผนวก ก โปรแกรมคำสั่งคอมพิวเตอร์

**โปรแกรม AB1-1 BAS ใช้สำหรับสร้างประชากรผู้สอบ ที่มีลักษณะการแจกแจงความสามารถของผู้สอบแบบปกติ จำนวน
 **1,200 คน

```

'Program Name : AB1-1.BAS
DECLARE SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
DECLARE FUNCTION Variance! (Var!())
DECLARE SUB NomalGen (Mean!, Variance!, Numvar&, Var!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
DECLARE SUB QuickSort (Low&, High&, SortArray!())
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION Skewness! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
*****
'   Program Name : AB1-1.BAS   *
*****
***** Simulation normal distribution *****
'$DYNAMIC
DIM SHARED Var!(1), Temp!(1), ObsFrq&(1), Upperbound!(1)
CONST pi = 3.141593
CLS
'
***** Generate Data VAR1() with theta +/- 3, SIZE =1200*****

DIM VAR1!(1 TO 1200)
PRINT "These variables come from Normal distribution generated by random no.13:-"
GENNormal 1200&, 3#, 13&, Var!(), Mn!, Var!
FOR I% = 1 TO 1200
  VAR1!(I%) = Var!(I%)
NEXT I%
PRINT "When select variable with theta +/- 3 and size 1200,"
PRINT "then these variables have parameters as follow:-"
PRINT "Mean   = ";
PRINT USING "###.####"; SMEAN!(VAR1!())
PRINT "Variance = ";
PRINT USING "###.####"; SVariance!(VAR1!())
PRINT "Kurtosis = ";
PRINT USING "###.####"; Kurtosis!(VAR1!(), Mn!, Var!)
PRINT "Skewness = ";
PRINT USING "###.####"; Skewness!(VAR1!(), Mn!, Var!)
OPEN "O", #1, "EXNEE.DAT"
FOR I% = 1 TO 1200
  PRINT #1, USING "###.####"; VAR1!(I%)
NEXT I%
CLOSE
PRINT "These variable has value as follow as:-"
FOR I% = 1 TO 1200 STEP 6
  PRINT USING "###.####"; VAR1!(I%); VAR1!(I% + 1); VAR1!(I% + 2); VAR1!(I% + 3); VAR1!(I% + 4); VAR1!(I% + 5)
NEXT I%

END
  
```

```

REM $STATIC
SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
  Prob# = 1 - Zscore#(UpDown#)
  Prob# = 1 - (2 * Prob#)
  SizeAll& = Size& / Prob#
  IF (SizeAll& MOD 2) <> 0 THEN SizeAll& = SizeAll& + 1
  REDIM Temp!(1 TO SizeAll&), Var!(1 TO Size&)
  Numvar& = SizeAll&
  Mn! = 0: Var! = 1
  RANDOMIZE Number&
  NormalGen Mn!, Var!, Numvar&, Temp!()
  PRINT "size", SizeAll&, "with :-"
  PRINT USING "Mean = ###.####", SMEAN!(Temp!())
  PRINT USING "Variance = ###.####", Variance!(Temp!())
  PRINT USING "Kurtosis = ###.####", Kurtosis!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
  PRINT USING "Skewness = ###.####", Skewness!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
  MMean! = SMEAN!(Temp!()); VVariance! = Variance!(Temp!())
  Round& = 0
  FOR I& = 1 TO SizeAll&
    IF (Temp!(I&) >= (-UpDown#)) AND (Temp!(I&) <= UpDown#) THEN
      Round& = Round& + 1
      Var!(Round&) = Temp!(I&)
      IF Round& = Size& THEN EXIT FOR
    END IF
  NEXT I&
  REDIM Temp!(1)
END SUB

```

```

FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
  Kur! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    Kur! = Kur! + (Var!(Number&) - Mean!) ^ 4
  NEXT Number&
  Kurtosis! = Kur! / (UBOUND(Var!) * (Sigma! ^ 4))
END FUNCTION

```

' Sub Routine for Generation of Normal Distribution By Box & Muller method

' MEAN! : Mean of Normal Distribution

' VARIANCE! : Variance of Normal Distribution

' Randomvar1! : Random variable no.1

' Randomvar2! : Random variable no.2

' Z1!,Z2! : Normal Random variables

' Store variabes = VAR!()

' Number o variabe = Numvar&

SUB NormalGen (Mean!, Variance!, Numvar&, Var!())

Number& = 0

DO

RandomVar1! = RND

RandomVar2! = RND

Z1! = Mean! + SQR(-2 * LOG(RandomVar1!)) * (COS(2 * pi * RandomVar2!)) * SQR(Variance!)

Number& = Number& + 1

Var!(Number&) = Z1!

IF Number& >= Numvar& THEN EXIT DO

Z2! = Mean! + SQR(-2 * LOG(RandomVar1!)) * (SIN(2 * pi * RandomVar2!)) * SQR(Variance!)

Number& = Number& + 1

Var!(Number&) = Z2!

```

    IF Number& >= Numvar& THEN EXIT DO
LOOP
END SUB

SUB QuickSort (Low&, High&, SortArray!())

    IF Low& < High& THEN

        ' Only two elements in this subdivision; swap them if they are out of
        ' order, then end recursive calls:
        IF High& - Low& = 1 THEN
            IF SortArray!(Low&) > SortArray!(High&) THEN
                SWAP SortArray!(Low&), SortArray!(High&)
            END IF
        ELSE

            ' Pick a pivot element at random, then move it to the end:
            RandIndex = RandInt&(Low&, High&)
            SWAP SortArray!(High&), SortArray!(RandIndex)
            Partition! = SortArray!(High&)
            DO

                ' Move in from both sides towards the pivot element:
                I& = Low&: J& = High&
                DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
                    I& = I& + 1
                LOOP
                DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
                    J& = J& - 1
                LOOP

                ' If we haven't reached the pivot element, it means that two
                ' elements on either side are out of order, so swap them:
                IF I& < J& THEN
                    SWAP SortArray!(I&), SortArray!(J&)
                END IF
            LOOP WHILE I& < J&

            ' Move the pivot element back to its proper place in the array:
            SWAP SortArray!(I&), SortArray!(High&)

            ' Recursively call the QuickSort procedure (pass the smaller
            ' subdivision first to use less stack space):
            IF (I& - Low&) < (High& - I&) THEN
                QuickSort Low&, I& - 1, SortArray!()
                QuickSort I& + 1, High&, SortArray!()
            ELSE
                QuickSort I& + 1, High&, SortArray!()
                QuickSort Low&, I& - 1, SortArray!()
            END IF
        END IF
    END IF
END SUB

```

```

FUNCTION RandInt& (lower&, Upper&) STATIC
  RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION Skewness! (Var!(), Mean!, Sigma!)
  SK! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    SK! = SK! + (Var!(Number&) - Mean!) ^ 3
  NEXT Number&
  Skewness! = SK! / (UBOUND(Var!) * (Sigma! ^ 3))
END FUNCTION

FUNCTION SMEAN! (Var!())
  m! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    m! = m! + Var!(Number&)
  NEXT Number&
  SMEAN! = m! / UBOUND(Var!)
END FUNCTION

FUNCTION SVariance! (Var!())
  m! = SMEAN!(Var!())
  S! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - m!) ^ 2
  NEXT Number&
  SVariance! = S! / (UBOUND(Var!) - 1)
END FUNCTION

SUB TestRandom (Numvar&, NoClass&, Chisquare!)
  REDIM Var!(1 TO Numvar&), ObsFrq&(1 TO NoClass&), Upperbound!(1 TO NoClass&)
  FOR I& = 1 TO Numvar&
    Var!(I&) = INT(RND(1) * 1000000!)
  NEXT I&
  QuickSort 1, Numvar&, Var!()
  ClassSize! = (Var!(Numvar&) - Var!(1)) / NoClass&
  FOR I& = 1 TO NoClass&
    Upperbound!(I&) = Var!(1) + I& * ClassSize!
  NEXT I&
  FOR I& = 1 TO Numvar&
    J& = 1
    DO WHILE Var!(I&) > Upperbound!(J&)
      J& = J& + 1
    LOOP
    ObsFrq&(J&) = ObsFrq&(J&) + 1
  NEXT I&
  ExpectedFrq! = Numvar& / NoClass&
  Chisquare! = 0
  FOR J& = 1 TO NoClass&
    PRINT ObsFrq&(J&), ExpectedFrq!,
    Chisquare! = Chisquare! + (((ObsFrq&(J&) - ExpectedFrq!) ^ 2) / ExpectedFrq!)
    PRINT ((ObsFrq&(J&) - ExpectedFrq!) ^ 2) / ExpectedFrq!
  NEXT J&
  PRINT Chisquare!
END SUB

```

```

FUNCTION Variance! (Var!())
  m! = SMEAN!(Var!())
  S! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - m!) ^ 2
  NEXT Number&
  Variance! = S! / UBOUND(Var!)
END FUNCTION

```

```

FUNCTION Zscore# (Z#)

```

' The formula used from Abramowitz & Stegun (1972) formula 2

```

IF Z# >= -14.14 AND Z# <= 14.14 THEN

```

```

  Y# = (.7071067812#) * Z#

```

```

ELSE

```

```

  Y# = 10

```

```

END IF

```

```

A1# = 7.052307839999999D-02

```

```

A2# = .0422820123#

```

```

A3# = .0092705272#

```

```

A4# = .0001520143#

```

```

A5# = .0002765672#

```

```

A6# = .0000430638#

```

```

Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) +
A6# * (Y# ^ 6)) ^ (-16)

```

```

END FUNCTION

```

โปรแกรม AB1-2.BAS ใช้สร้างประชากรข้อสอบ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยากของข้อสอบแบบปกติ

จำนวน 480 ข้อ

```

'Program Name :AB1-2.BAS
DECLARE SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
DECLARE FUNCTION Variance! (Var!())
DECLARE SUB NormalGen (Mean!, Variance!, Numvar&, Var!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION Skewness! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
*****
'*      Program Name :AB1-2.BAS
*****
'***** Simulation normal distribution *****
'$DYNAMIC
DIM SHARED Var!(1), Temp!(1), ObsFrq&(1), Upperbound!(1)
CONST pi = 3.141593
CLS
'***** Generate Data VAR2() with beta +/- 2, SIZE =480 *****
DIM VAR1!(1 TO 480)
PRINT "These variables come from Normal distribution generated by random no.19:-"
GENNormal 480&, 2#, 19&, Var!(), Mn!, Var!
FOR I% = 1 TO 480
    VAR1!(I%) = Var!(I%)
NEXT I%
PRINT "When select variable with beta +/- 2 and size 480 form Normal,"
PRINT "then these variables have parameters as follow:-"
PRINT "Mean = ";
PRINT USING "###.####"; SMEAN!(VAR1!())
PRINT "Variance = ";
PRINT USING "###.####"; SVariance!(VAR1!())
PRINT "Kurtosis = ";
PRINT USING "###.####"; Kurtosis!(VAR1!(), Mn!, Var!)
PRINT "Skewness = ";
PRINT USING "###.####"; Skewness!(VAR1!(), Mn!, Var!)

OPEN "O", #1, "ITEM-1.DAT"
FOR I% = 1 TO 480
    PRINT #1, USING "###.#####"; VAR1!(I%)
NEXT I%
CLOSE
PRINT "These variable has value as follow as:-"
FOR I% = 1 TO 480 STEP 6
    PRINT USING "###.##### "; VAR1!(I%); VAR1!(I% + 1); VAR1!(I% + 2); VAR1!(I% + 3); VAR1!(I% + 4); VAR1!(I% + 5)
NEXT I%
END

REM $STATIC
SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
    Prob# = 1 - Zscore#(UpDown#)
    Prob# = 1 - (2 * Prob#)
    SizeAll& = Size& / Prob#
    IF (SizeAll& MOD 2) <> 0 THEN SizeAll& = SizeAll& + 1

```

```

REDIM Temp!(1 TO SizeAll&), Var!(1 TO Size&), Temp1!(1 TO SizeAll&), Temp2!(1 TO SizeAll&)
Numvar& = SizeAll&
Mn! = 0: Var! = 1
RANDOMIZE Number&
NormalGen Mn!, Var!, Numvar&, Temp!()
PRINT "size"; SizeAll&; "with :-"
PRINT USING "Mean = ###.####"; SMEAN!(Temp!())
PRINT USING "Variance = ###.####"; Variance!(Temp!())
PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
PRINT USING "Skewness = ###.####"; Skewness!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
MMean! = SMEAN!(Temp!()); VVariance! = Variance!(Temp!())
Round& = 0
FOR I& = 1 TO SizeAll&
    IF (Temp!(I&) >= (-UpDown#)) AND (Temp!(I&) <= UpDown#) THEN
        Round& = Round& + 1
        Var!(Round&) = Temp!(I&)
        IF Round& = Size& THEN EXIT FOR
    END IF
NEXT I&

PRINT "When transfer negative Distribution form Normal,"
PRINT "by function is  $y = 2.345 + 1.1605091\# * X! - .2710708\# * X!^2 - .092819 * X!^3$ "
FOR I% = 1 TO SizeAll&
    X! = Temp!(I%)
    Temp1!(I%) = 2.345 + 1.1605091# * X! - .2710708# * X!^2 - .092819 * X!^3
NEXT I%
PRINT "size"; SizeAll&; "with :-"
PRINT USING "Mean = ###.####"; SMEAN!(Temp1!())
PRINT USING "Variance = ###.####"; Variance!(Temp1!())
PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp1!(), SMEAN!(Temp1!()), Variance!(Temp1!()))
PRINT USING "Skewness = ###.####"; Skewness!(Temp1!(), SMEAN!(Temp1!()), Variance!(Temp1!()))

PRINT "When transfer positive Distribution form Normal,"
PRINT "by function is  $y = 2.3268 + 1.14050961\# * X! + .2909708\# * X!^2 - .0886191 * X!^3$ "
FOR I% = 1 TO SizeAll&
    X! = Temp!(I%)
    Temp2!(I%) = 2.3268 + 1.14050961# * X! + .2909708# * X!^2 - .0886191 * X!^3
NEXT I%
PRINT "size"; SizeAll&; "with :-"
PRINT USING "Mean = ###.####"; SMEAN!(Temp2!())
PRINT USING "Variance = ###.####"; Variance!(Temp2!())
PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))
PRINT USING "Skewness = ###.####"; Skewness!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))

REDIM Temp!(1)
END SUB

FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
    Kur! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        Kur! = Kur! + (Var!(Number&) - Mean!)^4
    NEXT Number&
    Kurtosis! = Kur! / (UBOUND(Var!) * (Sigma! ^ 4))
END FUNCTION

```

```

' Sub Routine for Generation of Normal Distribution By Box & Muller method
' MEAN! : Mean of Normal Distribution
' VARIANCE! : Variance of Normal Distribution
' Randomvar1! : Random variable no.1
' Randomvar2! : Random variable no.2
' Z1!,Z2! : Normal Random variables
' Store variables = VAR!()
' Number o variabe = Numvar&
SUB NormalGen (Mean!, Variance!, Numvar&, Var!())
Number& = 0
DO
  RandomVar1! = RND
  RandomVar2! = RND
  Z1! = Mean! + SQR(-2 * LOG(RandomVar1!)) * (COS(2 * pi * RandomVar2!)) * SQR(Variance!)
  Number& = Number& + 1
  Var!(Number&) = Z1!
  IF Number& >= Numvar& THEN EXIT DO
  Z2! = Mean! + SQR(-2 * LOG(RandomVar1!)) * (SIN(2 * pi * RandomVar2!)) * SQR(Variance!)
  Number& = Number& + 1
  Var!(Number&) = Z2!
  IF Number& >= Numvar& THEN EXIT DO
LOOP
END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
  RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION Skewness! (Var!(), Mean!, Sigma!)
  SK! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    SK! = SK! + (Var!(Number&) - Mean!) ^ 3
  NEXT Number&
  Skewness! = SK! / (UBOUND(Var!) * (Sigma! ^ 3))
END FUNCTION

FUNCTION SMEAN! (Var!())
  m! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    m! = m! + Var!(Number&)
  NEXT Number&
  SMEAN! = m! / UBOUND(Var!)
END FUNCTION

FUNCTION SVariable! (Var!())
  m! = SMEAN!(Var!())
  S! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - m!) ^ 2
  NEXT Number&
  SVariable! = S! / (UBOUND(Var!) - 1)
END FUNCTION

FUNCTION Variance! (Var!())
  m! = SMEAN!(Var!())
  S! = 0

```

```

FOR Number& = 1 TO UBOUND(Var!)
  S! = S! + (Var!(Number&) - m!) ^ 2
NEXT Number&
Variance! = S! / UBOUND(Var!)
END FUNCTION

```

```

FUNCTION Zscore# (Z#)

```

```

' The formula used from Abramowitz & Stegun (1972) formula 2

```

```

IF Z# >= -14.14 AND Z# <= 14.14 THEN

```

```

  Y# = (.7071067812#) * Z#

```

```

ELSE

```

```

  Y# = 10

```

```

END IF

```

```

A1# = 7.052307839999999D-02

```

```

A2# = .0422820123#

```

```

A3# = .0092705272#

```

```

A4# = .0001520143#

```

```

A5# = .0002765672#

```

```

A6# = .0000430638#

```

```

Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) + A6# * (Y# ^ 6)) ^ (-16)

```

```

END FUNCTION

```

โปรแกรม NAB1-2.BAS ใช้สร้างประชากรข้อสอบ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยากของข้อสอบแบบ

แบบง่าย จำนวน 480 ข้อ

```

DECLARE SUB NormalGen (MMean!, VVariance!, Numvar&, Var!())
DECLARE SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
DECLARE FUNCTION Variance! (Var!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION Skewness! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
*****
'*      Program Name :NAB1-2.BAS
*****
*****      Simulation normal distribution      *****
'$DYNAMIC
A$ = COMMAND$
NNNN& = VAL(A$)
DIM SHARED Var!(1), Temp!(1), ObsFrq&(1), Upperbound!(1)
CONST pi = 3.141593
CLS

*****  Generate Data VAR2() with beta +/- 2, SIZE =480 *****
DIM VAR1!(1 TO 480)
PRINT "These variables come from Normal distribution generated by random no." + A$ + " :-"
GENNormal 480&, 2#, NNNN&, Var!(), Mn!, Var!
FOR I% = 1 TO 480
    VAR1!(I%) = Var!(I%)
NEXT I%

' Data with Negative skewness
OPEN "O", #1, "ITEM-2.DAT"
FOR I% = 1 TO 480
    PRINT #1, USING "###.#####"; VAR1!(I%)
NEXT I%
CLOSE
PRINT "When transfer Negative Distribution form Normal,"
PRINT "by function is  $y = 2.345 + 1.11605091X - 0.2710708X^2 - 0.092819X^3$ "
PRINT "then these variables have parameters as follow:-"
Mn! = SMEAN(VAR1!()); Var! = Variance!(VAR1!())
PRINT "Mean    = ";
PRINT USING "###.#####"; SMEAN!(VAR1!())
PRINT "Variance = ";
PRINT USING "###.#####"; SVariance!(VAR1!())
PRINT "Kurtosis = ";
PRINT USING "###.#####"; Kurtosis!(VAR1!(), Mn!, Var!)
PRINT "Skewness = ";
PRINT USING "###.#####"; Skewness!(VAR1!(), Mn!, Var!)

PRINT "These variable has value as follow as:-"
FOR I% = 1 TO 480 STEP 6
    PRINT USING "###.##### "; VAR1!(I%); VAR1!(I% + 1); VAR1!(I% + 2); VAR1!(I% + 3); VAR1!(I% + 4); VAR1!(I% + 5)
NEXT I%
END

```

```

REM $STATIC
SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
  Prob# = 1 - Zscore#(UpDown#)
  Prob# = 1 - (2 * Prob#)
  SizeAll& = Size& / Prob#
  FileNo% = FREEFILE
  OPEN "B", #FileNo%, "NLOOP.DAT"
  FOR KK% = 1 TO LOF(FileNo%) \ 4
    B$ = SPACE$(4)
    GET #FileNo%, , B$
    SizeAll& = SizeAll& + (Size& - VAL(B$))
  NEXT KK%
  IF (SizeAll& MOD 2) <> 0 THEN SizeAll& = SizeAll& + 1
  REDIM Temp1(1 TO SizeAll&), Var!(1 TO Size&), Temp1!(1 TO SizeAll&), Temp2!(1 TO SizeAll&)
  Numvar& = SizeAll&
  Mn! = 0: Var! = 1: Total% = 0
  RANDOMIZE Number&
  NormalGen Mn!, Var!, Numvar&, Temp1()
  PRINT "size"; SizeAll&; "with :-"
  PRINT USING "Mean = ###.####"; SMEAN!(Temp1())
  PRINT USING "Variance = ###.####"; Variance!(Temp1())
  PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp1(), SMEAN!(Temp1()), Variance!(Temp1()))
  PRINT USING "Skewness = ###.####"; Skewness!(Temp1(), SMEAN!(Temp1()), Variance!(Temp1()))
  MMean! = SMEAN!(Temp1()); VVariance! = Variance!(Temp1())
  PRINT "When transfer Negative Distribution form Normal,"
  PRINT "by function is  $y = 2.345 + 1.1605091 X - .2710708 X^2 - .092819 X^3$ "
  FOR I% = 1 TO SizeAll&
    X! = Temp1(I%)
    Temp2!(I%) = 2.345 + 1.1605091# * X! - .2710708# * X! ^ 2 - .092819 * X! ^ 3
  NEXT I%
  Round& = 0
  FOR I& = 1 TO SizeAll&
    IF (Temp2!(I&) >= (-UpDown#)) AND (Temp2!(I&) <= UpDown#) THEN
      Round& = Round& + 1
      Var!(Round&) = Temp2!(I&)
      IF Round& = Size& THEN EXIT FOR
    END IF
  NEXT I&
  PRINT "size"; SizeAll&; "with :-"
  PRINT USING "Mean = ###.####"; SMEAN!(Temp2!())
  PRINT USING "Variance = ###.####"; Variance!(Temp2!())
  PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))
  PRINT USING "Skewness = ###.####"; Skewness!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))
  IF Round& < Size& THEN
    SEEK #FileNo%, LOF(FileNo%) + 1
    B$ = STRING$(4, " ") + LTRIM$(RTRIM$(STR$(Round&)))
    B$ = RIGHT$(B$, 4)
    PUT #FileNo%, , B$
  END IF
  PRINT Round&
  CLOSE #FileNo%
  REDIM Temp1(1)
END SUB

```

```
FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
```

```
    Kur! = 0
```

```
    FOR Number& = 1 TO UBOUND(Var!)
```

```
        Kur! = Kur! + (Var!(Number&) - Mean!) ^ 4
```

```
    NEXT Number&
```

```
    Kurtosis! = Kur! / (UBOUND(Var!) * (Sigma! ^ 4))
```

```
END FUNCTION
```

```
' Sub Routine for Generation of Normal Distribution By Box & Muller method
```

```
' MMean! : Mean of Normal Distribution
```

```
' VVariance! : Variance of Normal Distribution
```

```
' Randomvar1! : Random variable no.1
```

```
' Randomvar2! : Random variable no.2
```

```
' Z1!,Z2! : Normal Random variables
```

```
' Store variables = VAR!()
```

```
' Number o variabe = Numvar&
```

```
SUB NormalGen (MMean!, VVariance!, Numvar&, Var!())
```

```
    Number& = 0
```

```
    DO
```

```
        RandomVar1! = RND
```

```
        RandomVar2! = RND
```

```
        Z1! = MMean! + SQR(-2 * LOG(RandomVar1!)) * (COS(2 * pi * RandomVar2!)) * SQR(VVariance!)
```

```
        Number& = Number& + 1
```

```
        Var!(Number&) = Z1!
```

```
        IF Number& >= Numvar& THEN EXIT DO
```

```
        Z2! = MMean! + SQR(-2 * LOG(RandomVar1!)) * (SIN(2 * pi * RandomVar2!)) * SQR(VVariance!)
```

```
        Number& = Number& + 1
```

```
        Var!(Number&) = Z2!
```

```
        IF Number& >= Numvar& THEN EXIT DO
```

```
    LOOP
```

```
END SUB
```

```
FUNCTION RandInt& (lower&, Upper&) STATIC
```

```
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
```

```
END FUNCTION
```

```
FUNCTION Skewness! (Var!(), Mean!, Sigma!)
```

```
    SK! = 0
```

```
    FOR Number& = 1 TO UBOUND(Var!)
```

```
        SK! = SK! + (Var!(Number&) - Mean!) ^ 3
```

```
    NEXT Number&
```

```
    Skewness! = SK! / (UBOUND(Var!) * (Sigma! ^ 3))
```

```
END FUNCTION
```

```
FUNCTION SMEAN! (Var!())
```

```
    m! = 0
```

```
    FOR Number& = 1 TO UBOUND(Var!)
```

```
        m! = m! + Var!(Number&)
```

```
    NEXT Number&
```

```
    SMEAN! = m! / UBOUND(Var!)
```

```
END FUNCTION
```

```
FUNCTION SVariance! (Var!())
```

```
    m! = SMEAN!(Var!())
```

```
    S! = 0
```

```

FOR Number& = 1 TO UBOUND(Var!)
  S! = S! + (Var!(Number&) - m!) ^ 2
NEXT Number&
SVariance! = S! / (UBOUND(Var!) - 1)
END FUNCTION

```

```

FUNCTION Variance! (Var!())
  m! = SMEAN!(Var!())
  S! = 0
  FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - m!) ^ 2
  NEXT Number&
  Variance! = S! / UBOUND(Var!)
END FUNCTION

```

```

FUNCTION Zscore# (Z#)
' The formula used from Abramowitz & Stegun (1972) formula 2

IF Z# >= -14.14 AND Z# <= 14.14 THEN
  Y# = (.7071067812#) * Z#
ELSE
  Y# = 10
END IF
A1# = 7.052307839999999D-02
A2# = .0422820123#
A3# = .0092705272#
A4# = .0001520143#
A5# = .0002765672#
A6# = .0000430638#
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) + A6# * (Y# ^ 6)) ^ (-16)
END FUNCTION

```

โปรแกรม AB1-3.BAS ใช้คำนวณหาค่า A_{ij} และ X_{ij} ระหว่างประชากรผู้สอบและประชากรข้อสอบทั้ง 3 กรณี

```

*****
'      Program Name :AB1-3.BAS
*****
TYPE Fstr
  Row AS INTEGER
  Col AS INTEGER
  Prob AS SINGLE
  Aij AS SINGLE
  X AS STRING * 1
END TYPE
TYPE FFF
  X AS STRING * 1
END TYPE
DIM SHARED Theta!(1 TO 1200), Beta!(1 TO 480)
DIM SHARED File AS Fstr, File2 AS FFF
CLS
OPEN "I", #1, "EXNEE.DAT"
I% = 0
DO WHILE NOT EOF(1)
  LINE INPUT #1, NN$
  I% = I% + 1
  Theta!(I%) = VAL(NN$)
LOOP
CLOSE
OPEN "I", #1, "ITEM-1.DAT"
J% = 0
DO WHILE NOT EOF(1)
  LINE INPUT #1, NN$
  J% = J% + 1
  Beta!(J%) = VAL(NN$)
  PRINT Beta!(J%)
LOOP
CLOSE
PRINT "Examinee-Normal Distribution & Item-Normal Distribution"
PRINT "-----"
PRINT "Examinee Theta   Item   Beta   Probability   Aij   Xij"
PRINT "-----"
OPEN "R", #1, "P1-1.DAT", LEN(File)
RANDOMIZE 12
Lno% = 0
FOR I% = 1 TO 1200
  FOR J% = 1 TO 480
    File.Row = I%
    PRINT TAB(2); USING "#,###"; I%;
    PRINT TAB(10); USING "###.####"; Theta!(I%);
    File.Col = J%
    PRINT TAB(20); USING "#,###"; J%;
    PRINT TAB(30); USING "###.####"; Beta!(J%);
    P1! = (1 / (1 + (2.71828) ^ (-1.7 * (Theta!(I%) - Beta!(J%)))))
    PRINT TAB(43); USING "###.####"; P1!;
    File.Prob = P1!
    Aij! = RND
    PRINT TAB(56); USING "###.####"; Aij!;
  
```

```

File.Aij = Aij!
IF P1! < Aij! THEN
    XX$ = "0"
ELSE
    XX$ = "1"
END IF
File.X = XX$
PRINT TAB(70); XX$
Lno% = Lno% + 1
IF Lno% >= 55 THEN
    Lno% = 0
    PRINT CHR$(12)
    PRINT "Examinee-Normal Distribution & Item-Normal Distribution"
    PRINT "-----"
    PRINT "Examinee Theta   Item   Beta   Probability   Aij   Xij"
    PRINT "-----"
END IF
PUT #1, LOF(1) \ LEN(File) + 1, File
NEXT J%
NEXT I%
CLOSE

OPEN "I", #1, "ITEM-2.DAT"
J% = 0
DO WHILE NOT EOF(1)
    LINE INPUT #1, NN$
    J% = J% + 1
    Beta!(J%) = VAL(NN$)
LOOP
CLOSE

PRINT "Examinee-Normal Distribution & Item-Negative Distribution"
PRINT "-----"
PRINT "Examinee Theta   Item   Beta   Probability   Aij   Xij"
PRINT "-----"
OPEN "R", #1, "P1-2.DAT", LEN(File)
Lno% = 0
FOR I% = 1 TO 1200
    FOR J% = 1 TO 480
        File.Row = I%
        PRINT TAB(2); USING "#,###"; I%;
        PRINT TAB(10); USING "###.####"; Theta!(I%);
        File.Col = J%
        PRINT TAB(20); USING "#,###"; J%;
        PRINT TAB(30); USING "###.####"; Beta!(J%);
        P1! = (1 / (1 + (2.71828) ^ (-1.7 * (Theta!(I%) - Beta!(J%)))))
        PRINT TAB(43); USING "###.####"; P1!;
        File.Prob = P1!
        Aij! = RND
        PRINT TAB(56); USING "###.####"; Aij!;
        File.Aij = Aij!
        IF P1! < Aij! THEN
            XX$ = "0"
        ELSE
            XX$ = "1"

```

```

END IF
PRINT TAB(70); XX$
Lno% = Lno% + 1
IF Lno% >= 55 THEN
    Lno% = 0
    PRINT CHR$(12)
    PRINT "Examinee-Normal Distribution & Item-Negative Distribution"
    PRINT "-----"
    PRINT "Examinee Theta   Item   Beta   Probability   Aij   Xij"
    PRINT "-----"
END IF
File.X = XX$
PUT #1, LOF(1) \ LEN(File) + 1, File
NEXT J%
NEXT I%
CLOSE

OPEN "I", #1, "ITEM-3.DAT"
J% = 0
DO WHILE NOT EOF(1)
    LINE INPUT #1, NN$
    J% = J% + 1
    Beta!(J%) = VAL(NN$)
LOOP
CLOSE
END
PRINT "Examinee-Normal Distribution & Item-Positive Distribution"
PRINT "-----"
PRINT "Examinee Theta   Item   Beta   Probability   Aij   Xij"
PRINT "-----"
OPEN "R", #1, "P1-3.DAT", LEN(File)
Lno% = 0
FOR I% = 1 TO 1200
    FOR J% = 1 TO 480
        File.Row = I%
        PRINT TAB(2); USING "#,###"; I%;
        PRINT TAB(10); USING "##.####"; Theta!(I%);
        File.Col = J%
        PRINT TAB(20); USING "#,###"; J%;
        PRINT TAB(30); USING "##.####"; Beta!(J%);
        P1! = (1 / (1 + (2.71828) ^ (-1.7 * (Theta!(I%) - Beta!(J%)))))
        PRINT TAB(43); USING "##.####"; P1!;
        File.Prob = P1!
        Aij! = RND
        File.Aij = Aij!
        PRINT TAB(56); USING "##.####"; Aij!;
        IF P1! < Aij! THEN
            XX$ = "0"
        ELSE
            XX$ = "1"
        END IF
        PRINT TAB(70); XX$
        Lno% = Lno% + 1
    IF Lno% >= 55 THEN
        Lno% = 0
    END IF

```

```

PRINT CHR$(12)
PRINT "Examinee-Normal Distribution & Item-Positive Distribution"
PRINT "-----"
PRINT "Examinee Theta   Iteme   Beta   Probability   Aij   Xij"
PRINT "-----"
END IF
File.X = XX$
PUT #1, LOF(1) \ LEN(File) + 1, File
NEXT J%
NEXT I%
CLOSE
OPEN "R", #1, "P1-1.DAT", LEN(File)
OPEN "R", #2, "X1-1.DAT", LEN(File2)
FOR I& = 1 TO LOF(1) \ LEN(File)
  GET #1, I&, File
  File2.X = File.X
  PUT #2, LOF(2) \ LEN(File2) + 1, File2
NEXT I&
CLOSE
OPEN "R", #1, "P1-2.DAT", LEN(File)
OPEN "R", #2, "X1-2.DAT", LEN(File2)
FOR I& = 1 TO LOF(1) \ LEN(File)
  GET #1, I&, File
  File2.X = File.X
  PUT #2, LOF(2) \ LEN(File2) + 1, File2
NEXT I&
CLOSE
OPEN "R", #1, "P1-3.DAT", LEN(File)
OPEN "R", #2, "X1-3.DAT", LEN(File2)
FOR I& = 1 TO LOF(1) \ LEN(File)
  GET #1, I&, File
  File2.X = File.X
  PUT #2, LOF(2) \ LEN(File2) + 1, File2
NEXT I&
CLOSE
END

```

"โปรแกรม AB1-4.BAS สำหรับคำนวณคะแนนของผู้สอบแต่ละคนในการทำข้อสอบทั้ง 3 กรณี

' Program Name : AB1-4.BAS

***** Calculating & printing parameters *****

TYPE Fstr

Row AS INTEGER

Col AS INTEGER

Prob AS SINGLE

Uni AS SINGLE

X AS STRING * 1

END TYPE

DIM SHARED File AS Fstr

DIM Score1!(1 TO 1200), Score2!(1 TO 1200), Score3!(1 TO 1200)

CLS

PRINT "Reading data..."

Sum1! = 0: Sum2! = 0: Sum3! = 0

OPEN "R", #1, "P1-1.DAT", LEN(File)

FOR I& = 1 TO LOF(1) \ LEN(File)

GET #1, I&, File

Row% = File.Row

Score1!(Row%) = Score1!(Row%) + VAL(File.X)

Sum1! = Sum1! + VAL(File.X)

NEXT I&

CLOSE

OPEN "R", #1, "P1-2.DAT", LEN(File)

FOR I& = 1 TO LOF(1) \ LEN(File)

GET #1, I&, File

Row% = File.Row

Score2!(Row%) = Score2!(Row%) + VAL(File.X)

Sum2! = Sum2! + VAL(File.X)

NEXT I&

CLOSE

OPEN "R", #1, "P1-3.DAT", LEN(File)

FOR I& = 1 TO LOF(1) \ LEN(File)

GET #1, I&, File

Row% = File.Row

Score3!(Row%) = Score3!(Row%) + VAL(File.X)

Sum3! = Sum3! + VAL(File.X)

NEXT I&

CLOSE

OPEN "O", #1, "MEAN.DAT"

PRINT "calculating parameters..."

PRINT "Parameters of score from 3 Items Distributions:-"

PRINT "-----"

PRINT

Mn1! = (Sum1! / (1200! * 480!))

Var1! = Mn1! - (Mn1!) ^ 2

PRINT "Score (Item with NORMAL Distribution)with :-"

PRINT USING "Mean = ##.###.###"; Mn1!

PRINT USING "Variance = ##.###.###"; Var1!

```

PRINT USING "S.D. = ###,###.####"; SQR(Var1!)
PRINT #1, USING "####.####"; Mn1!;
PRINT #1, USING "####.####"; Var1!
PRINT
PRINT
Mn2! = (Sum2! / (1200! * 480!))
Var2! = Mn2! - (Mn2!) ^ 2
PRINT "Score (Item with Negative Skewness Distribution) .....with :-"
PRINT USING "Mean = ###,###.####"; Mn2!
PRINT USING "Variance = ###,###.####"; Var2!
PRINT USING "S.D. = ###,###.####"; SQR(Var2!)
PRINT #1, USING "####.####"; Mn2!;
PRINT #1, USING "####.####"; Var2!
PRINT
PRINT
Mn3! = (Sum3! / (1200! * 480!))
Var3! = Mn3! - (Mn3!) ^ 2
PRINT "Score (Item with Positive Skewness Distribution) .....with :-"
PRINT USING "Mean = ###,###.####"; Mn3!
PRINT USING "Variance = ###,###.####"; Var3!
PRINT USING "S.D. = ###,###.####"; SQR(Var3!)
PRINT #1, USING "####.####"; Mn3!;
PRINT #1, USING "####.####"; Var3!
PRINT
PRINT
PRINT "-----"
END

```

โปรแกรม PAB1-2 BAS ใช้สร้างประชากรข้อสอบ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยากของข้อสอบแบบ

แบบวา จำนวน 480 ข้อ

```

DECLARE SUB NormalGen (MMean!, VVariance!, Numvar&, Var!())
DECLARE SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
DECLARE FUNCTION Variance! (Var!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION Skewness! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
*****
'*      Program Name :PAB1-2.BAS
*****
***** Simulation normal distribution *****
'$DYNAMIC
A$ = COMMAND$
NNNN& = VAL(A$)
DIM SHARED Var!(1), Temp!(1), ObsFrq&(1), Upperbound!(1)
CONST pi = 3.141593
CLS

***** Generate Data VAR2() with beta +/- 2, SIZE =480 *****
DIM VAR1!(1 TO 480)
PRINT "These variables come from Normal distribution generated by random no." + A$ + " :-"
GENNormal 480&, 2#, NNNN&, Var!(), Mn!, Var!
FOR I% = 1 TO 480
    VAR1!(I%) = Var!(I%)
NEXT I%

' Data with Positive skewness
OPEN "O", #1, "ITEM-3.DAT"
FOR I% = 1 TO 480
    PRINT #1, USING "###.#####", VAR1!(I%)
NEXT I%
CLOSE
PRINT "When transfer positive Distribution form Normal,"
PRINT "by function is  $y = -2.3268 + 1.16050961X + 0.2909708X^2 - 0.0886191X^3$ "
PRINT "then these variables have parameters as follow:-"
Mn! = SMEAN(VAR1!()); Var! = Variance!(VAR1!())
PRINT "Mean = ";
PRINT USING "###.#####", SMEAN!(VAR1!())
PRINT "Variance = ";
PRINT USING "###.#####", SVariance!(VAR1!())
PRINT "Kurtosis = ";
PRINT USING "###.#####", Kurtosis!(VAR1!(), Mn!, Var!)
PRINT "Skewness = ";
PRINT USING "###.#####", Skewness!(VAR1!(), Mn!, Var!)

PRINT "These variable has value as follow as:-"
FOR I% = 1 TO 480 STEP 6
    PRINT USING "###.##### "; VAR1!(I%); VAR1!(I% + 1); VAR1!(I% + 2); VAR1!(I% + 3); VAR1!(I% + 4); VAR1!(I% + 5)
NEXT I%
END

```

```

REM $STATIC
SUB GENNormal (Size&, UpDown#, Number&, Var!(), MMean!, VVariance!)
  Prob# = 1 - Zscore#(UpDown#)
  Prob# = 1 - (2 * Prob#)
  SizeAll& = Size& / Prob#
  FileNo% = FREEFILE
  OPEN "B", #FileNo%, "PLOOP.DAT"
  FOR KK% = 1 TO LOF(FileNo%) \ 4
    B$ = SPACE$(4)
    GET #FileNo%, , B$
    SizeAll& = SizeAll& + (Size& - VAL(B$))
  NEXT KK%
  IF (SizeAll& MOD 2) <> 0 THEN SizeAll& = SizeAll& + 1
  REDIM Temp!(1 TO SizeAll&), Var!(1 TO Size&), Temp1!(1 TO SizeAll&), Temp2!(1 TO SizeAll&)
  Numvar& = SizeAll&
  Mn! = 0: Var! = 1: Total% = 0
  RANDOMIZE Number&
  NormalGen Mn!, Var!, Numvar&, Temp!()
  PRINT "size"; SizeAll&; "with :-"
  PRINT USING "Mean = ###.####"; SMEAN!(Temp!())
  PRINT USING "Variance = ###.####"; Variance!(Temp!())
  PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
  PRINT USING "Skewness = ###.####"; Skewness!(Temp!(), SMEAN!(Temp!()), Variance!(Temp!()))
  MMean! = SMEAN!(Temp!()); VVariance! = Variance!(Temp!())

  PRINT "When transfer positive Distribution form Normal,"
  PRINT "by function is  $y = -2.3268 + 1.14050961# X + .2909708 X^2 - .0886191 X^3$ "
  FOR I% = 1 TO SizeAll&
    X! = Temp!(I%)
    Temp2!(I%) = -2.3268 + 1.16050961# * X! + .2909708# * X! ^ 2 - .0886191 * X! ^ 3
  NEXT I%
  Round& = 0
  FOR I& = 1 TO SizeAll&
    IF (Temp2!(I&) >= (-UpDown#)) AND (Temp2!(I&) <= UpDown#) THEN
      Round& = Round& + 1
      Var!(Round&) = Temp2!(I&)
      IF Round& = Size& THEN EXIT FOR
    END IF
  NEXT I&
  PRINT "size"; SizeAll&; "with :-"
  PRINT USING "Mean = ###.####"; SMEAN!(Temp2!())
  PRINT USING "Variance = ###.####"; Variance!(Temp2!())
  PRINT USING "Kurtosis = ###.####"; Kurtosis!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))
  PRINT USING "Skewness = ###.####"; Skewness!(Temp2!(), SMEAN!(Temp2!()), Variance!(Temp2!()))
  IF Round& < Size& THEN
    SEEK #FileNo%, LOF(FileNo%) + 1
    B$ = STRING$(4, " ") + LTRIM$(RTRIM$(STR$(Round&)))
    B$ = RIGHT$(B$, 4)
    PUT #FileNo%, , B$
  END IF
  PRINT Round&
  CLOSE #FileNo%
  REDIM Temp!(1)
END SUB

```

```

FUNCTION Kurtosis! (Var!(), Mean!, Sigma!)
    Kur! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        Kur! = Kur! + (Var!(Number&) - Mean!) ^ 4
    NEXT Number&
    Kurtosis! = Kur! / (UBOUND(Var!) * (Sigma! ^ 4))
END FUNCTION

' Sub Routine for Generation of Normal Distribution By Box & Muller method
' MMean! : Mean of Normal Distribution
' VVariance! : Variance of Normal Distribution
' Randomvar1! : Random variable no.1
' Randomvar2! : Random variable no.2
' Z1!,Z2! : Normal Random variables
' Store variables = VAR!()
' Number o variabe = Numvar&
SUB NormalGen (MMean!, VVariance!, Numvar&, Var!())
    Number& = 0
    DO
        RandomVar1! = RND
        RandomVar2! = RND
        Z1! = MMean! + SQR(-2 * LOG(RandomVar1!)) * (COS(2 * pi * RandomVar2!)) * SQR(VVariance!)
        Number& = Number& + 1
        Var!(Number&) = Z1!
        IF Number& >= Numvar& THEN EXIT DO
        Z2! = MMean! + SQR(-2 * LOG(RandomVar1!)) * (SIN(2 * pi * RandomVar2!)) * SQR(VVariance!)
        Number& = Number& + 1
        Var!(Number&) = Z2!
        IF Number& >= Numvar& THEN EXIT DO
    LOOP
END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION Skewness! (Var!(), Mean!, Sigma!)
    SK! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        SK! = SK! + (Var!(Number&) - Mean!) ^ 3
    NEXT Number&
    Skewness! = SK! / (UBOUND(Var!) * (Sigma! ^ 3))
END FUNCTION

FUNCTION SMEAN! (Var!())
    m! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        m! = m! + Var!(Number&)
    NEXT Number&
    SMEAN! = m! / UBOUND(Var!)
END FUNCTION

FUNCTION SVariance! (Var!())
    m! = SMEAN!(Var!())
    S! = 0
    FOR Number& = 1 TO UBOUND(Var!)

```

```

    S! = S! + (Var!(Number&) - m!) ^ 2
NEXT Number&
SVariance! = S! / (UBOUND(Var!) - 1)
END FUNCTION

```

```

FUNCTION Variance! (Var!())
    m! = SMEAN!(Var!())
    S! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        S! = S! + (Var!(Number&) - m!) ^ 2
    NEXT Number&
    Variance! = S! / UBOUND(Var!)
END FUNCTION

```

```

FUNCTION Zscore# (Z#)
' The formula used from Abramowitz & Stegun (1972) formula 2

IF Z# >= -14.14 AND Z# <= 14.14 THEN
    Y# = (.7071067812#) * Z#
ELSE
    Y# = 10
END IF
A1# = 7.052307839999999D-02
A2# = .0422820123#
A3# = .0092705272#
A4# = .0001520143#
A5# = .0002765672#
A6# = .0000430638#
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) + A6# * (Y# ^ 6)) ^ (-16)
END FUNCTION

```

“โปรแกรม P_AB2.BAS ใช้คำนวณข้อมูลเตรียมไว้สำหรับการทำงานของโปรแกรม AB2-1,AB2-2,AB2-3.BAS ต่อไป

CLS

,

TYPE Fstr

Row AS INTEGER

Col AS INTEGER

Prob AS SINGLE

Aij AS SINGLE

X AS STRING * 1

END TYPE

TYPE Fstr2

Mark AS SINGLE

END TYPE

DIM SHARED File AS Fstr, File2 AS Fstr2

,

***** Normal Distribution *****

,

OPEN "R", #1, "P1-1.DAT", LEN(File)

OPEN "R", #2, "MP1-1.DAT", LEN(File2)

FOR I% = 1 TO 1200

Score& = 0

RecordStart& = (I% - 1) * 480& + 1

FOR Record& = RecordStart& TO RecordStart& + 479

GET #1, Record&, File

Score& = Score& + VAL(File.X)

NEXT Record&

File2.Mark = Score&

PUT #2, I%, File2

NEXT I%

CLOSE

***** Negative Skewness Distribution *****

,

OPEN "R", #1, "P1-2.DAT", LEN(File)

OPEN "R", #2, "MP1-2.DAT", LEN(File2)

FOR I% = 1 TO 1200

Score& = 0

RecordStart& = (I% - 1) * 480& + 1

FOR Record& = RecordStart& TO RecordStart& + 479

GET #1, Record&, File

Score& = Score& + VAL(File.X)

NEXT Record&

File2.Mark = Score&

PUT #2, I%, File2

NEXT I%

CLOSE

***** Positive skewness Distribution *****

,

OPEN "R", #1, "P1-3.DAT", LEN(File)

OPEN "R", #2, "MP1-3.DAT", LEN(File2)

FOR I% = 1 TO 1200

Score& = 0

RecordStart& = (I% - 1) * 480& + 1

FOR Record& = RecordStart& TO RecordStart& + 479

GET #1, Record&, File

```
Score& = Score& + VAL(File.X)  
NEXT Record&  
File2.Mark = Score&  
PUT #2, I%, File2  
NEXT I%  
CLOSE  
END
```

โปรแกรม AB2-1 BAS ใช้สำหรับกระบวนการสุ่มแบบง่าย กรณีข้อสอบมีลักษณะการแจกแจงค่าความยากของข้อสอบ

'แบบปกติ

' Program Name :AB2-1.bas

DECLARE FUNCTION SumScore2! (VAR!Q)

DECLARE FUNCTION SumScore! (VAR!Q)

DECLARE FUNCTION RandInt& (lower&, Upper&)

***** SRS PROCESS *****

' \$DYNAMIC

CLS

,

TYPE Fstr

Mark AS SINGLE

END TYPE

TYPE Fstr2

Round AS INTEGER

Group AS INTEGER

SumX AS SINGLE

SumX2 AS SINGLE

END TYPE

DIM SHARED File AS Fstr, File2 AS Fstr2

CLS

,

***** Normal Distribution *****

,

' n=600 by 1,000x for the test Normal Distribution

,

OPEN "R", #1, "MP1-1.DAT", LEN(File)

OPEN "R", #2, "S6001-1.DAT", LEN(File2)

DIM Check%(1 TO 1200)

DIM Score1!(1 TO 600)

DIM Score2!(1 TO 600)

RANDOMIZE 1

FOR Round% = 1 TO 1000

REDIM Check%(1 TO 1200)

REDIM Score1!(1 TO 600)

REDIM Score2!(1 TO 600)

' Group-1 n=600

FOR I% = 1 TO 600

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 1

NEXT I%

' Group-2 n=600

FOR I% = 1 TO 1200

IF Check%(I%) <> 1 THEN

Check%(I%) = 2

END IF

NEXT I%

IdCheckGroup1% = 0: IdCheckGroup2% = 0

FOR I% = 1 TO 1200

IF Check%(I%) = 1 THEN

IdCheckGroup1% = IdCheckGroup1% + 1

GET #1, I%, File

```

        Score1!(IdCheckGroup1%) = File.Mark
    ELSE
        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    END IF
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 2, File2
LOCATE 1, 1: PRINT "SRS 600 Round "; Round%
NEXT Round%
CLOSE #2

```

' n=300 by 1,000x for the test Normal Distribution

```

OPEN "R", #2, "S3001-1.DAT", LEN(File2)
REDIM Check%(1 TO 1200)
DIM Score3!(1 TO 300)
DIM Score4!(1 TO 300)
RANDOMIZE 2
FOR Round% = 1 TO 1000
    REDIM Check%(1 TO 1200)
    REDIM Score1!(1 TO 300)
    REDIM Score2!(1 TO 300)
    REDIM Score3!(1 TO 300)
    REDIM Score4!(1 TO 300)
    ' Group-1 n=300
    FOR I% = 1 TO 300
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 1
        NEXT I%
    ' Group-2 n=300
    FOR I% = 1 TO 300
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 2
        NEXT I%
    
```

```

' Group-3 n=300
FOR I% = 1 TO 300
    DO
        A& = RandInt&(1, 1200)
        LOOP UNTIL Check%(A&) = 0
        Check%(A&) = 3
    NEXT I%

' Group-4 n=300
FOR I% = 1 TO 1200
    IF Check%(I%) = 0 THEN
        Check%(I%) = 4
    END IF
NEXT I%

IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0: IdCheckGroup4% = 0
FOR I% = 1 TO 1200
    IF Check%(I%) = 1 THEN
        IdCheckGroup1% = IdCheckGroup1% + 1
        GET #1, I%, File
        Score1!(IdCheckGroup1%) = File.Mark
    ELSEIF Check%(I%) = 2 THEN
        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    ELSEIF Check%(I%) = 3 THEN
        IdCheckGroup3% = IdCheckGroup3% + 1
        GET #1, I%, File
        Score3!(IdCheckGroup3%) = File.Mark
    ELSE
        IdCheckGroup4% = IdCheckGroup4% + 1
        GET #1, I%, File
        Score4!(IdCheckGroup4%) = File.Mark
    END IF
NEXT I%

File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!

```

```

PUT #2, (Round% - 1) * 4 + 3, File2
File2.Round = Round%
File2.Group = 4
X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 4, File2
LOCATE 1, 1: PRINT "SRS 300 Round "; Round%
NEXT Round%
CLOSE #2

```

```

' n=200 by 1,000x for the test Normal Distribution

```

```

OPEN "R", #2, "S2001-1.DAT", LEN(File2)
REDIM Check%(1 TO 1200)
DIM Score5!(1 TO 200)
DIM Score6!(1 TO 200)
RANDOMIZE 3
FOR Round% = 1 TO 1000
  REDIM Check%(1 TO 1200)
  REDIM Score1!(1 TO 200)
  REDIM Score2!(1 TO 200)
  REDIM Score3!(1 TO 200)
  REDIM Score4!(1 TO 200)
  REDIM Score5!(1 TO 200)
  REDIM Score6!(1 TO 200)
  ' Group-1 n=200
  FOR I% = 1 TO 200
    DO
      A& = RandInt&(1, 1200)
      LOOP UNTIL Check%(A&) = 0
      Check%(A&) = 1
    NEXT I%
  ' Group-2 n=200
  FOR I% = 1 TO 200
    DO
      A& = RandInt&(1, 1200)
      LOOP UNTIL Check%(A&) = 0
      Check%(A&) = 2
    NEXT I%
  ' Group-3 n=200
  FOR I% = 1 TO 200
    DO
      A& = RandInt&(1, 1200)
      LOOP UNTIL Check%(A&) = 0
      Check%(A&) = 3
    NEXT I%
  ' Group-4 n=200
  FOR I% = 1 TO 200
    DO
      A& = RandInt&(1, 1200)
      LOOP UNTIL Check%(A&) = 0
    NEXT I%
  NEXT Round%

```

```

    Check%(A&) = 4
NEXT I%
' Group-5 n=200
FOR I% = 1 TO 200
    DO
        A& = RandInt&(1, 1200)
        LOOP UNTIL Check%(A&) = 0
        Check%(A&) = 5
    NEXT I%

' Group-6 n=200
FOR I% = 1 TO 1200
    IF Check%(I%) = 0 THEN
        Check%(I%) = 6
    END IF
NEXT I%
IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0
IdCheckGroup4% = 0: IdCheckGroup5% = 0: IdCheckGroup6% = 0
FOR I% = 1 TO 1200
    SELECT CASE Check%(I%)
    CASE IS = 1
        IdCheckGroup1% = IdCheckGroup1% + 1
        GET #1, I%, File
        Score1!(IdCheckGroup1%) = File.Mark
    CASE IS = 2
        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    CASE IS = 3
        IdCheckGroup3% = IdCheckGroup3% + 1
        GET #1, I%, File
        Score3!(IdCheckGroup3%) = File.Mark
    CASE IS = 4
        IdCheckGroup4% = IdCheckGroup4% + 1
        GET #1, I%, File
        Score4!(IdCheckGroup4%) = File.Mark
    CASE IS = 5
        IdCheckGroup5% = IdCheckGroup5% + 1
        GET #1, I%, File
        Score5!(IdCheckGroup5%) = File.Mark
    CASE ELSE
        IdCheckGroup6% = IdCheckGroup6% + 1
        GET #1, I%, File
        Score6!(IdCheckGroup6%) = File.Mark
    END SELECT
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 1, File2
File2.Round = Round%

```

```

File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 3, File2
File2.Round = Round%
File2.Group = 4
X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 4, File2
File2.Round = Round%
File2.Group = 5
X! = SumScore!(Score5!())
File2.SumX = X!
X! = SumScore2!(Score5!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 5, File2
File2.Round = Round%
File2.Group = 6
X! = SumScore!(Score6!())
File2.SumX = X!
X! = SumScore2!(Score6!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 6, File2
LOCATE 1, 1: PRINT "SRS 200 Round "; Round%
NEXT Round%
CLOSE #2, #1
END

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())

IF Low& < High& THEN

    ' Only two elements in this subdivision; swap them if they are out of
    ' order, then end recursive calls:
    IF High& - Low& = 1 THEN
        IF SortArray!(Low&) > SortArray!(High&) THEN
            SWAP SortArray!(Low&), SortArray!(High&)
        END IF
    ELSE

        ' Pick a pivot element at random, then move it to the end:
        RandIndex = RandInt&(Low&, High&)

```

```

SWAP SortArray!(High&), SortArray!(RandIndex)
Partition! = SortArray!(High&)
DO
  ' Move in from both sides towards the pivot element:
  I& = Low&: J& = High&
  DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
    I& = I& + 1
  LOOP
  DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
    J& = J& - 1
  LOOP
  ' If we haven't reached the pivot element, it means that two
  ' elements on either side are out of order, so swap them:
  IF I& < J& THEN
    SWAP SortArray!(I&), SortArray!(J&)
  END IF
  LOOP WHILE I& < J&
  ' Move the pivot element back to its proper place in the array:
  SWAP SortArray(I&), SortArray(High&)
  ' Recursively call the QuickSort procedure (pass the smaller
  ' subdivision first to use less stack space):
  IF (I& - Low&) < (High& - I&) THEN
    QuickSort Low&, I& - 1, SortArray!()
    QuickSort I& + 1, High&, SortArray!()
  ELSE
    QuickSort I& + 1, High&, SortArray!()
    QuickSort Low&, I& - 1, SortArray!()
  END IF
END IF
END IF
END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
  RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION SumScore! (VAR!())
  Number& = UBOUND(VAR!)
  Sum! = 0
  FOR I& = 1 TO Number&
    Sum! = Sum! + VAR!(I&)
  NEXT I&
  SumScore! = Sum!
END FUNCTION

FUNCTION SumScore2! (VAR!())
  Number& = UBOUND(VAR!)
  Sum! = 0
  FOR I& = 1 TO Number&
    Sum! = Sum! + VAR!(I&) ^ 2
  NEXT I&
  SumScore2! = Sum!
END FUNCTION

```

โปรแกรม AB2-2.BAS ใช้สำหรับกระบวนการสุ่มแบบง่าย กรณีข้อสอบมีลักษณะการแจกแจงค่าความยากของข้อสอบ
'แบบง่าย'

```
*****
'      Program Name :AB2-2.bas
*****
DECLARE FUNCTION SumScore2! (VAR!())
DECLARE FUNCTION SumScore! (VAR!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
***** SRS PROCESS *****
'$DYNAMIC
CLS
TYPE Fstr
    Mark AS SINGLE
END TYPE
TYPE Fstr2
    Round AS INTEGER
    Group AS INTEGER
    SumX AS SINGLE
    SumX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File2 AS Fstr2
CLS
'
***** Negative Skewness Distribution *****
' n=600 by 1,000x for the test Negative Skewness Distribution
'
OPEN "R", #1, "MP1-2.DAT", LEN(File)
OPEN "R", #2, "S6001-2.DAT", LEN(File2)
DIM Check%(1 TO 1200)
DIM Score1!(1 TO 600)
DIM Score2!(1 TO 600)
RANDOMIZE 4
FOR Round% = 1 TO 1000
    REDIM Check%(1 TO 1200)
    REDIM Score1!(1 TO 600)
    REDIM Score2!(1 TO 600)
    ' Group-1 n=600
    FOR I% = 1 TO 600
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 1
        NEXT I%
    ' Group-2 n=600
    FOR I% = 1 TO 1200
        IF Check%(I%) <> 1 THEN
            Check%(I%) = 2
        END IF
    NEXT I%
    IdCheckGroup1% = 0: IdCheckGroup2% = 0
    FOR I% = 1 TO 1200
        IF Check%(I%) = 1 THEN
            IdCheckGroup1% = IdCheckGroup1% + 1
        GET #1, I%, File
```

```

        Score2!(IdCheckGroup2%) = File.Mark
    END IF
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 2, File2
LOCATE 1, 1: PRINT "SRS 600 Round "; Round%
NEXT Round%
CLOSE #2
' n=300 by 1,000x for the test Negative Skewness Distribution

OPEN "R", #2, "S3001-2.DAT", LEN(File2)
REDIM Check%(1 TO 1200)
DIM Score3!(1 TO 300)
DIM Score4!(1 TO 300)
RANDOMIZE 5
FOR Round% = 1 TO 1000
    REDIM Check%(1 TO 1200)
    REDIM Score1!(1 TO 300)
    REDIM Score2!(1 TO 300)
    REDIM Score3!(1 TO 300)
    REDIM Score4!(1 TO 300)
    ' Group-1 n=300
    FOR I% = 1 TO 300
        DO
            A% = RandInt&(1, 1200)
            LOOP UNTIL Check%(A%) = 0
            Check%(A%) = 1
        NEXT I%
    ' Group-2 n=300
    FOR I% = 1 TO 300
        DO
            A% = RandInt&(1, 1200)
            LOOP UNTIL Check%(A%) = 0
            Check%(A%) = 2
        NEXT I%

    ' Group-3 n=300
    FOR I% = 1 TO 300
        DO
            A% = RandInt&(1, 1200)
            LOOP UNTIL Check%(A%) = 0
            Check%(A%) = 3
        NEXT I%

```

```

' Group-4 n=300
FOR I% = 1 TO 1200
    IF Check%(I%) = 0 THEN
        Check%(I%) = 4
    END IF
NEXT I%
IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0: IdCheckGroup4% = 0
FOR I% = 1 TO 1200
    IF Check%(I%) = 1 THEN
        IdCheckGroup1% = IdCheckGroup1% + 1
        GET #1, I%, File
        Score1!(IdCheckGroup1%) = File.Mark
    ELSEIF Check%(I%) = 2 THEN
        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    ELSEIF Check%(I%) = 3 THEN
        IdCheckGroup3% = IdCheckGroup3% + 1
        GET #1, I%, File
        Score3!(IdCheckGroup3%) = File.Mark
    ELSE
        IdCheckGroup4% = IdCheckGroup4% + 1
        GET #1, I%, File
        Score4!(IdCheckGroup4%) = File.Mark
    END IF
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 3, File2
File2.Round = Round%
File2.Group = 4
X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 4, File2
LOCATE 1, 1: PRINT "SRS 300 Round "; Round%
NEXT Round%

```

CLOSE #2

'n=200 by 1,000x for the test Negative Skewness Distribution

OPEN "R", #2, "S2001-2.DAT", LEN(File2)

REDIM Check%(1 TO 1200)

DIM Score5!(1 TO 200)

DIM Score6!(1 TO 200)

RANDOMIZE 6

FOR Round% = 1 TO 1000

REDIM Check%(1 TO 1200)

REDIM Score1!(1 TO 200)

REDIM Score2!(1 TO 200)

REDIM Score3!(1 TO 200)

REDIM Score4!(1 TO 200)

REDIM Score5!(1 TO 200)

REDIM Score6!(1 TO 200)

' Group-1 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 1

NEXT I%

' Group-2 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 2

NEXT I%

' Group-3 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 3

NEXT I%

' Group-4 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 4

NEXT I%

' Group-5 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 5

NEXT I%

' Group-6 n=200

FOR I% = 1 TO 1200

IF Check%(I%) = 0 THEN

```

    Check%(I%) = 6
  END IF
NEXT I%
IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0
IdCheckGroup4% = 0: IdCheckGroup5% = 0: IdCheckGroup6% = 0
FOR I% = 1 TO 1200
  SELECT CASE Check%(I%)
    CASE IS = 1
      IdCheckGroup1% = IdCheckGroup1% + 1
      GET #1, I%, File
      Score1!(IdCheckGroup1%) = File.Mark
    CASE IS = 2
      IdCheckGroup2% = IdCheckGroup2% + 1
      GET #1, I%, File
      Score2!(IdCheckGroup2%) = File.Mark
    CASE IS = 3
      IdCheckGroup3% = IdCheckGroup3% + 1
      GET #1, I%, File
      Score3!(IdCheckGroup3%) = File.Mark
    CASE IS = 4
      IdCheckGroup4% = IdCheckGroup4% + 1
      GET #1, I%, File
      Score4!(IdCheckGroup4%) = File.Mark
    CASE IS = 5
      IdCheckGroup5% = IdCheckGroup5% + 1
      GET #1, I%, File
      Score5!(IdCheckGroup5%) = File.Mark
    CASE ELSE
      IdCheckGroup6% = IdCheckGroup6% + 1
      GET #1, I%, File
      Score6!(IdCheckGroup6%) = File.Mark
  END SELECT
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 3, File2
File2.Round = Round%
File2.Group = 4

```

```

X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 4, File2
File2.Round = Round%
File2.Group = 5
X! = SumScore!(Score5!())
File2.SumX = X!
X! = SumScore2!(Score5!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 5, File2
File2.Round = Round%
File2.Group = 6
X! = SumScore!(Score6!())
File2.SumX = X!
X! = SumScore2!(Score6!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 6, File2
LOCATE 1, 1: PRINT "SRS 200 Round "; Round%
NEXT Round%
CLOSE #2, #1
END

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())

IF Low& < High& THEN

    ' Only two elements in this subdivision; swap them if they are out of
    ' order, then end recursive calls:
    IF High& - Low& = 1 THEN
        IF SortArray!(Low&) > SortArray!(High&) THEN
            SWAP SortArray!(Low&), SortArray!(High&)
        END IF
    ELSE
        ' Pick a pivot element at random, then move it to the end:
        RandIndex = RandInt&(Low&, High&)
        SWAP SortArray!(High&), SortArray!(RandIndex)
        Partition! = SortArray!(High&)
        DO

            ' Move in from both sides towards the pivot element:
            I& = Low&: J& = High&
            DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
                I& = I& + 1
            LOOP
            DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
                J& = J& - 1
            LOOP

            ' If we haven't reached the pivot element, it means that two
            ' elements on either side are out of order, so swap them:
            IF I& < J& THEN
                SWAP SortArray!(I&), SortArray!(J&)
            
```

```

    END IF
    LOOP WHILE I& < J&

    ' Move the pivot element back to its proper place in the array:
    SWAP SortArray(I&), SortArray(High&)

    ' Recursively call the QuickSort procedure (pass the smaller
    ' subdivision first to use less stack space):
    IF (I& - Low&) < (High& - I&) THEN
        QuickSort Low&, I& - 1, SortArray!()
        QuickSort I& + 1, High&, SortArray!()
    ELSE
        QuickSort I& + 1, High&, SortArray!()
        QuickSort Low&, I& - 1, SortArray!()
    END IF
END IF
END IF
END IF

END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION SumScore! (VAR!())
    Number& = UBOUND(VAR!)
    Sum! = 0
    FOR I& = 1 TO Number&
        Sum! = Sum! + VAR!(I&)
    NEXT I&
    SumScore! = Sum!
END FUNCTION

FUNCTION SumScore2! (VAR!())
    Number& = UBOUND(VAR!)
    Sum! = 0
    FOR I& = 1 TO Number&
        Sum! = Sum! + VAR!(I&) ^ 2
    NEXT I&
    SumScore2! = Sum!
END FUNCTION

```

โปรแกรม AB2-3.BAS ใช้สำหรับกระบวนการสุ่มแบบง่าย กรณีข้อสอบมีลักษณะการแจกแจงค่าความยาวของข้อสอบ

'แบบเบ้ขวา

```

*****
'      Program Name :AB2-3.bas
*****
DECLARE FUNCTION SumScore2! (VAR!())
DECLARE FUNCTION SumScore! (VAR!())
DECLARE FUNCTION RandInt& (lower&, Upper&)
***** SRS PROCESS *****
'$DYNAMIC
CLS
'
TYPE Fstr
    Mark AS SINGLE
END TYPE
TYPE Fstr2
    Round AS INTEGER
    Group AS INTEGER
    SumX AS SINGLE
    SumX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File2 AS Fstr2
CLS
'
***** Positive Skewness Distribution *****
'
'n=600 by 1,000x for the test Positive Skewness Distribution
'
OPEN "R", #1, "MP1-3.DAT", LEN(File)
OPEN "R", #2, "S6001-3.DAT", LEN(File2)
DIM Check%(1 TO 1200)
DIM Score1!(1 TO 600)
DIM Score2!(1 TO 600)
RANDOMIZE 4
FOR Round% = 1 TO 1000
    REDIM Check%(1 TO 1200)
    REDIM Score1!(1 TO 600)
    REDIM Score2!(1 TO 600)
    ' Group-1 n=600
    FOR I% = 1 TO 600
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 1
        NEXT I%
    ' Group-2 n=600
    FOR I% = 1 TO 1200
        IF Check%(I%) <> 1 THEN
            Check%(I%) = 2
        END IF
    NEXT I%
    IdCheckGroup1% = 0: IdCheckGroup2% = 0
    FOR I% = 1 TO 1200
        IF Check%(I%) = 1 THEN
            IdCheckGroup1% = IdCheckGroup1% + 1
            GET #1, I%, File
            Score1!(IdCheckGroup1%) = File.Mark
        ELSE

```

```

        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    END IF
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 2 + 2, File2
LOCATE 1, 1: PRINT "SRS 600 Round "; Round%
NEXT Round%
CLOSE #2
' n=300 by 1,000x for the test Positive Skewness Distribution
OPEN "R", #2, "S3001-3.DAT", LEN(File2)
REDIM Check%(1 TO 1200)
DIM Score3!(1 TO 300)
DIM Score4!(1 TO 300)
RANDOMIZE 5
FOR Round% = 1 TO 1000
    REDIM Check%(1 TO 1200)
    REDIM Score1!(1 TO 300)
    REDIM Score2!(1 TO 300)
    REDIM Score3!(1 TO 300)
    REDIM Score4!(1 TO 300)
    ' Group-1 n=300
    FOR I% = 1 TO 300
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 1
        DO
    NEXT I%
    ' Group-2 n=300
    FOR I% = 1 TO 300
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 2
        DO
    NEXT I%
    ' Group-3 n=300
    FOR I% = 1 TO 300
        DO
            A& = RandInt&(1, 1200)
            LOOP UNTIL Check%(A&) = 0
            Check%(A&) = 3
        DO
    NEXT I%

```

```

' Group-4 n=300
FOR I% = 1 TO 1200
  IF Check%(I%) = 0 THEN
    Check%(I%) = 4
  END IF
NEXT I%
IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0: IdCheckGroup4% = 0
FOR I% = 1 TO 1200
  IF Check%(I%) = 1 THEN
    IdCheckGroup1% = IdCheckGroup1% + 1
    GET #1, I%, File
    Score1!(IdCheckGroup1%) = File.Mark
  ELSEIF Check%(I%) = 2 THEN
    IdCheckGroup2% = IdCheckGroup2% + 1
    GET #1, I%, File
    Score2!(IdCheckGroup2%) = File.Mark
  ELSEIF Check%(I%) = 3 THEN
    IdCheckGroup3% = IdCheckGroup3% + 1
    GET #1, I%, File
    Score3!(IdCheckGroup3%) = File.Mark
  ELSE
    IdCheckGroup4% = IdCheckGroup4% + 1
    GET #1, I%, File
    Score4!(IdCheckGroup4%) = File.Mark
  END IF
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 3, File2
File2.Round = Round%
File2.Group = 4
X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 4 + 4, File2
LOCATE 1, 1: PRINT "SRS 300 Round "; Round%

```

NEXT Round%

CLOSE #2

'n=200 by 1,000x for the test Positive Skewness Distribution

OPEN "R", #2, "S2001-3.DAT", LEN(File2)

REDIM Check%(1 TO 1200)

DIM Score5!(1 TO 200)

DIM Score6!(1 TO 200)

RANDOMIZE 6

FOR Round% = 1 TO 1000

REDIM Check%(1 TO 1200)

REDIM Score1!(1 TO 200)

REDIM Score2!(1 TO 200)

REDIM Score3!(1 TO 200)

REDIM Score4!(1 TO 200)

REDIM Score5!(1 TO 200)

REDIM Score6!(1 TO 200)

' Group-1 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 1

NEXT I%

' Group-2 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 2

NEXT I%

' Group-3 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 3

NEXT I%

' Group-4 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 4

NEXT I%

' Group-5 n=200

FOR I% = 1 TO 200

DO

A& = RandInt&(1, 1200)

LOOP UNTIL Check%(A&) = 0

Check%(A&) = 5

NEXT I%

' Group-6 n=200

FOR I% = 1 TO 1200

```

IF Check%(I%) = 0 THEN
    Check%(I%) = 6
END IF
NEXT I%
IdCheckGroup1% = 0: IdCheckGroup2% = 0: IdCheckGroup3% = 0
IdCheckGroup4% = 0: IdCheckGroup5% = 0: IdCheckGroup6% = 0
FOR I% = 1 TO 1200
    SELECT CASE Check%(I%)
    CASE IS = 1
        IdCheckGroup1% = IdCheckGroup1% + 1
        GET #1, I%, File
        Score1!(IdCheckGroup1%) = File.Mark
    CASE IS = 2
        IdCheckGroup2% = IdCheckGroup2% + 1
        GET #1, I%, File
        Score2!(IdCheckGroup2%) = File.Mark
    CASE IS = 3
        IdCheckGroup3% = IdCheckGroup3% + 1
        GET #1, I%, File
        Score3!(IdCheckGroup3%) = File.Mark
    CASE IS = 4
        IdCheckGroup4% = IdCheckGroup4% + 1
        GET #1, I%, File
        Score4!(IdCheckGroup4%) = File.Mark
    CASE IS = 5
        IdCheckGroup5% = IdCheckGroup5% + 1
        GET #1, I%, File
        Score5!(IdCheckGroup5%) = File.Mark
    CASE ELSE
        IdCheckGroup6% = IdCheckGroup6% + 1
        GET #1, I%, File
        Score6!(IdCheckGroup6%) = File.Mark
    END SELECT
NEXT I%
File2.Round = Round%
File2.Group = 1
X! = SumScore!(Score1!())
File2.SumX = X!
X! = SumScore2!(Score1!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 1, File2
File2.Round = Round%
File2.Group = 2
X! = SumScore!(Score2!())
File2.SumX = X!
X! = SumScore2!(Score2!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 2, File2
File2.Round = Round%
File2.Group = 3
X! = SumScore!(Score3!())
File2.SumX = X!
X! = SumScore2!(Score3!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 3, File2
File2.Round = Round%

```

```

File2.Group = 4
X! = SumScore!(Score4!())
File2.SumX = X!
X! = SumScore2!(Score4!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 4, File2
File2.Round = Round%
File2.Group = 5
X! = SumScore!(Score5!())
File2.SumX = X!
X! = SumScore2!(Score5!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 5, File2
File2.Round = Round%
File2.Group = 6
X! = SumScore!(Score6!())
File2.SumX = X!
X! = SumScore2!(Score6!())
File2.SumX2 = X!
PUT #2, (Round% - 1) * 6 + 6, File2
LOCATE 1, 1: PRINT "SRS 200 Round ", Round%
NEXT Round%
CLOSE #2, #1
END

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())

IF Low& < High& THEN

    ' Only two elements in this subdivision; swap them if they are out of
    ' order, then end recursive calls:
    IF High& - Low& = 1 THEN
        IF SortArray!(Low&) > SortArray!(High&) THEN
            SWAP SortArray!(Low&), SortArray!(High&)
        END IF
    ELSE

        ' Pick a pivot element at random, then move it to the end:
        RandIndex = RandInt&(Low&, High&)
        SWAP SortArray!(High&), SortArray!(RandIndex)
        Partition! = SortArray!(High&)
        DO

            ' Move in from both sides towards the pivot element:
            I& = Low&: J& = High&
            DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
                I& = I& + 1
            LOOP
            DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
                J& = J& - 1
            LOOP

            ' If we haven't reached the pivot element, it means that two
            ' elements on either side are out of order, so swap them:
            IF I& < J& THEN

```

```

        SWAP SortArray!(I&), SortArray!(J&)
    END IF
    LOOP WHILE I& < J&

    ' Move the pivot element back to its proper place in the array:
    SWAP SortArray(I&), SortArray(High&)

    ' Recursively call the QuickSort procedure (pass the smaller
    ' subdivision first to use less stack space):
    IF (I& - Low&) < (High& - I&) THEN
        QuickSort Low&, I& - 1, SortArray!()
        QuickSort I& + 1, High&, SortArray!()
    ELSE
        QuickSort I& + 1, High&, SortArray!()
        QuickSort Low&, I& - 1, SortArray!()
    END IF
END IF
END IF

END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

FUNCTION SumScore! (VAR!())
    Number& = UBOUND(VAR!)
    Sum! = 0
    FOR I& = 1 TO Number&
        Sum! = Sum! + VAR!(I&)
    NEXT I&
    SumScore! = Sum!
END FUNCTION

FUNCTION SumScore2! (VAR!())
    Number& = UBOUND(VAR!)
    Sum! = 0
    FOR I& = 1 TO Number&
        Sum! = Sum! + VAR!(I&) ^ 2
    NEXT I&
    SumScore2! = Sum!
END FUNCTION

```

โปรแกรม AB3-1.BAS ใช้สำหรับกระบวนการสุ่มแบบเมตริกซ์เชิงพหุคูณ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยาก

ของข้อสอบแบบปกติ

```
'
      AB3-1.bas
DECLARE FUNCTION RandInt& (lower&, Upper&)
***** MMS PROCESS *****
'$DYNAMIC
CLS
'
TYPE Fstr
  Mark AS SINGLE
END TYPE
TYPE Fstr1
  X AS STRING * 1
END TYPE
TYPE Fstr2
  Round AS INTEGER
  Group AS INTEGER
  SUME AS SINGLE
  SUME2 AS SINGLE
  SUM1 AS SINGLE
  SUMI2 AS SINGLE
  SUMX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File1 AS Fstr1, File2 AS Fstr2
CLS
PRINT "Calculating....."
'
***** Normal Distribution *****
N& = 1200: M& = 480
'
'k=2 by 1,000x for the test Normal Distribution
'
ns& = 600: ms& = 240
OPEN "R", #1, "MMS600-1.DAT", LEN(File2)
DIM Check1%(1 TO N&)
DIM Check2%(1 TO M&)
DIM EScore!(1 TO N&)
DIM IScore!(1 TO M&)
RANDOMIZE 10
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 2) / 2 + 1
FOR Round% = 1 TO RR%

  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  'Group-1 n=600
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  'Group-2 n=600
  FOR I% = 1 TO N&
    IF Check1%(I%) <> 1 THEN
      Check1%(I%) = 2
    
```

```

END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=240
FOR I% = 1 TO ms&
DO
  A& = RandInt&(1, M&)
  LOOP UNTIL Check2%(A&) = 0
  Check2%(A&) = 1
NEXT I%
' Group-2 m=240
FOR I% = 1 TO M&
  IF Check2%(I%) <> 1 THEN
    Check2%(I%) = 2
  END IF
NEXT I%
NEXT Round%
FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 600: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  ' Group-1 n=600
  FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 1
  NEXT I%
  ' Group-2 n=600
  FOR I% = 1 TO N&
    IF Check1%(I%) <> 1 THEN
      Check1%(I%) = 2
    END IF
  NEXT I%
  ***** Sampling Examinees *****
  REDIM Check2%(1 TO M&)

  ' Group-1 m=240
  FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
  ' Group-2 m=240
  FOR I% = 1 TO M&
    IF Check2%(I%) <> 1 THEN
      Check2%(I%) = 2
    END IF
  NEXT I%

```

'Print Samples & Calculate Item Score & Examinee Score

```

REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0
OPEN "R", #2, "X1-1.DAT", LEN(File1)
FOR I% = 1 TO N&
  LOCATE 1, 70: PRINT I%
  RecordStart& = (I% - 1) * M& + 1
  FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    SELECT CASE Check1%(I%)
      CASE 1
        IF Check2%(Record& - RecordStart& + 1) = 1 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
        END IF
      CASE ELSE
        IF Check2%(Record& - RecordStart& + 1) = 2 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
        END IF
      END SELECT
    NEXT Record&
  NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
FOR I% = 1 TO N&
  SELECT CASE Check1%(I%)
    CASE 1
      SE1! = SE1! + EScore!(I%)
      SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
      SE2! = SE2! + EScore!(I%)
      SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    END SELECT
  NEXT I%
FOR J% = 1 TO M&
  SELECT CASE Check2%(J%)
    CASE 1
      SI1! = SI1! + IScore!(J%)
      SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
    CASE ELSE
      SI2! = SI2! + IScore!(J%)
      SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
    END SELECT
  NEXT J%

File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!

```

```

File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 2 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 2 + 2, File2
CLOSE #1
OPEN "R", #1, "MMS600-1.DAT", LEN(File2)
NEXT Round%
CLOSE #1

'
***** Normal Distribution *****
N& = 1200: M& = 480
'
' k=4 by 1,000x for the test Normal Distribution
'
ns& = 300: ms& = 120
OPEN "R", #1, "MMS300-1.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 11
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 4) / 4 + 1

FOR Round% = 1 TO RR%
  REDIM Check1%(1 TO N&)
  ' Group-1 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=300

```

```

FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 4
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)
' Group-1 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=120
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 4
  END IF
NEXT I%
NEXT Round%

FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 300: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  ' Group-1 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  NEXT Round%

```

```

NEXT I%
' Group-3 n=300
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 3
  NEXT I%
' Group-4 n=300
FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 4
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=120
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 4
  END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0
OPEN "R", #2, "X1-1.DAT", LEN(File1)
FOR I% = 1 TO N&
  LOCATE 1, 70: PRINT I%
  RecordStart& = (I% - 1) * M& + 1
  FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    LOCATE 2, 70: PRINT (Record& - RecordStart& + 1); ":", Check2%(Record& - RecordStart& + 1)
    SELECT CASE Check1%(I%)

```

```

CASE 1
  IF Check2%(Record& - RecordStart& + 1) = 1 THEN
    GET #2, Record&, File1
    EScore!(I%) = EScore!(I%) + VAL(File1.X)
    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
    SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
  END IF
CASE 2
  IF Check2%(Record& - RecordStart& + 1) = 2 THEN
    GET #2, Record&, File1
    EScore!(I%) = EScore!(I%) + VAL(File1.X)
    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
    SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
  END IF
CASE 3
  IF Check2%(Record& - RecordStart& + 1) = 3 THEN
    GET #2, Record&, File1
    EScore!(I%) = EScore!(I%) + VAL(File1.X)
    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
    SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
  END IF
CASE ELSE
  IF Check2%(Record& - RecordStart& + 1) = 4 THEN
    GET #2, Record&, File1
    EScore!(I%) = EScore!(I%) + VAL(File1.X)
    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
    SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
  END IF
END SELECT
NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
FOR I% = 1 TO N&
  SELECT CASE Check1%(I%)
  CASE 1
    SE1! = SE1! + EScore!(I%)
    SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
  CASE 2
    SE2! = SE2! + EScore!(I%)
    SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
  CASE 3
    SE3! = SE3! + EScore!(I%)
    SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
  CASE ELSE
    SE4! = SE4! + EScore!(I%)
    SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
  END SELECT
NEXT I%
FOR J% = 1 TO M&
  SELECT CASE Check2%(J%)
  CASE 1

```

```

      SI1! = SI1! + IScore!(J%)
      SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
CASE 2
      SI2! = SI2! + IScore!(J%)
      SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
CASE 3
      SI3! = SI3! + IScore!(J%)
      SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
CASE ELSE
      SI4! = SI4! + IScore!(J%)
      SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
END SELECT
NEXT J%
File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 4 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 4 + 2, File2
File2.Round = Round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (Round% - 1) * 4 + 3, File2
File2.Round = Round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (Round% - 1) * 4 + 4, File2
CLOSE #1
OPEN "R", #1, "MMS300-1.DAT", LEN(File2)
NEXT Round%
CLOSE #1

```

***** Normal Distribution *****

N& = 1200: M& = 480

'k=6 by 1,000x for the test Normal Distribution

```

ns& = 200: ns& = 80
OPEN "R", #1, "MMS200-1.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 12
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 6) / 6 + 1

FOR Round% = 1 TO RR%
  ' Group-1 n=200
  REDIM Check1%(1 TO N&)
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 4
    NEXT I%
  ' Group-5 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 5
    NEXT I%
  ' Group-6 n=200
  FOR I% = 1 TO N&
    IF Check1%(I%) = 0 THEN
      Check1%(I%) = 6
    END IF
  NEXT I%
  REDIM Check2%(1 TO M&)

```

```

ns& = 200: ms& = 80
OPEN "R", #1, "MMS200-1.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 12
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 6) / 6 + 1

```

```

FOR Round% = 1 TO RR%
  ' Group-1 n=200
  REDIM Check1%(1 TO N&)
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 4
    NEXT I%
  ' Group-5 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 5
    NEXT I%
  ' Group-6 n=200
  FOR I% = 1 TO N&
    IF Check1%(I%) = 0 THEN
      Check1%(I%) = 6
    END IF
  NEXT I%
  REDIM Check2%(1 TO M&)

```

```

' Group-1 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 4
  NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 5
  NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(C%) = 6
  END IF
NEXT I%
NEXT Round%
FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 200: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

' Group-1 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 1
  NEXT I%

```

```

' Group-2 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 2
  NEXT I%
' Group-3 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 3
  NEXT I%
' Group-4 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 4
  NEXT I%
' Group-5 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 5
  NEXT I%
' Group-6 n=200
FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 6
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)

```

```

    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 4
    NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 5
    NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
    IF Check2%(I%) = 0 THEN
        Check2%(I%) = 6
    END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0: SumX5Sqr! = 0: SumX6Sqr! = 0
OPEN "R", #2, "X1-1.DAT", LEN(File1)
FOR I% = 1 TO N&
    LOCATE 1, 70: PRINT I%
    RecordStart& = (I% - 1) * M& + 1
    FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
        SELECT CASE Check1%(I%)
        CASE 1
            IF Check2%(Record& - RecordStart& + 1) = 1 THEN
                GET #2, Record&, File1
                EScore!(I%) = EScore!(I%) + VAL(File1.X)
                IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
            END IF
        CASE 2
            IF Check2%(Record& - RecordStart& + 1) = 2 THEN
                GET #2, Record&, File1
                EScore!(I%) = EScore!(I%) + VAL(File1.X)
                IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
            END IF
        CASE 3
            IF Check2%(Record& - RecordStart& + 1) = 3 THEN
                GET #2, Record&, File1
                EScore!(I%) = EScore!(I%) + VAL(File1.X)
                IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
            END IF
        CASE 4
            IF Check2%(Record& - RecordStart& + 1) = 4 THEN

```

```

        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
    END IF
CASE 5
    IF Check2%(Record& - RecordStart& + 1) = 5 THEN
        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX5Sqr! = SumX5Sqr! + (VAL(File1.X) ^ 2)
    END IF
CASE ELSE
    IF Check2%(Record& - RecordStart& + 1) = 6 THEN
        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX6Sqr! = SumX6Sqr! + (VAL(File1.X) ^ 2)
    END IF
END SELECT
NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
SE5! = 0: SE5Sqr! = 0: SI5! = 0: SI5Sqr! = 0
SE6! = 0: SE6Sqr! = 0: SI6! = 0: SI6Sqr! = 0
FOR I% = 1 TO N&
    SELECT CASE Check1%(I%)
    CASE 1
        SE1! = SE1! + EScore!(I%)
        SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE 2
        SE2! = SE2! + EScore!(I%)
        SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    CASE 3
        SE3! = SE3! + EScore!(I%)
        SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
    CASE 4
        SE4! = SE4! + EScore!(I%)
        SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
    CASE 5
        SE5! = SE5! + EScore!(I%)
        SE5Sqr! = SE5Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
        SE6! = SE6! + EScore!(I%)
        SE6Sqr! = SE6Sqr! + (EScore!(I%) ^ 2)
    END SELECT
NEXT I%
FOR J% = 1 TO M&
    SELECT CASE Check2%(J%)
    CASE 1
        SI1! = SI1! + IScore!(J%)

```

```

    SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
CASE 2
    SI2! = SI2! + IScore!(J%)
    SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
CASE 3
    SI3! = SI3! + IScore!(J%)
    SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
CASE 4
    SI4! = SI4! + IScore!(J%)
    SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
CASE 5
    SI5! = SI5! + IScore!(J%)
    SI5Sqr! = SI5Sqr! + (IScore!(J%) ^ 2)
CASE ELSE
    SI6! = SI6! + IScore!(J%)
    SI6Sqr! = SI6Sqr! + (IScore!(J%) ^ 2)
END SELECT
NEXT J%
File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 6 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 6 + 2, File2
File2.Round = Round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (Round% - 1) * 6 + 3, File2
File2.Round = Round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (Round% - 1) * 6 + 4, File2
File2.Round = Round%
File2.Group = 5
File2.SUME = SE5!
File2.SUME2 = SE5Sqr!
File2.SUMI = SI5!

```

```

File2.SUMI2 = SI5Sqr!
File2.SUMX2 = SumX5Sqr!
PUT #1, (Round% - 1) * 6 + 5, File2
File2.Round = Round%
File2.Group = 6
File2.SUME = SE6!
File2.SUME2 = SE6Sqr!
File2.SUMI = SI6!
File2.SUMI2 = SI6Sqr!
File2.SUMX2 = SumX6Sqr!
PUT #1, (Round% - 1) * 6 + 6, File2
CLOSE #1
OPEN "R", #1, "MMS200-1.DAT", LEN(File2)
NEXT Round%
CLOSE #1
END

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())
  IF Low& < High& THEN
    ' Only two elements in this subdivision; swap them if they are out of
    ' order, then end recursive calls:
    IF High& - Low& = 1 THEN
      IF SortArray!(Low&) > SortArray!(High&) THEN
        SWAP SortArray!(Low&), SortArray!(High&)
      END IF
    ELSE
      ' Pick a pivot element at random, then move it to the end:
      RandIndex = RandInt&(Low&, High&)
      SWAP SortArray!(High&), SortArray!(RandIndex)
      Partition! = SortArray!(High&)
      DO
        ' Move in from both sides towards the pivot element:
        I& = Low&: J& = High&
        DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
          I& = I& + 1
        LOOP
        DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
          J& = J& - 1
        LOOP
        ' If we haven't reached the pivot element, it means that two
        ' elements on either side are out of order, so swap them:
        IF I& < J& THEN
          SWAP SortArray!(I&), SortArray!(J&)
        END IF
      LOOP WHILE I& < J&
      ' Move the pivot element back to its proper place in the array:
      SWAP SortArray!(I&), SortArray!(High&)
      ' Recursively call the QuickSort procedure (pass the smaller
      ' subdivision first to use less stack space):
      IF (I& - Low&) < (High& - I&) THEN
        QuickSort Low&, I& - 1, SortArray!()
        QuickSort I& + 1, High&, SortArray!()
      ELSE
        QuickSort I& + 1, High&, SortArray!()
      END IF
    END IF
  END IF
END SUB

```

```
        QuickSort Low&, I& - 1, SortArray!()  
    END IF  
END IF  
END IF  
  
END SUB  
  
FUNCTION RandInt& (lower&, Upper&) STATIC  
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&  
END FUNCTION
```

โปรแกรม AB3-2 BAS ใช้สำหรับกระบวนการสุ่มแบบเมตริกซ์เชิงพหุคูณ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยาก

'ของข้อสอบแบบเป็นซ้าย

```

1  * * * * * AB3-2.bas * * * * *
DECLARE FUNCTION RandInt& (lower&, Upper&)
' * * * * * MMS PROCESS * * * * *
'$DYNAMIC
CLS

TYPE Fstr
    Mark AS SINGLE
END TYPE
TYPE Fstr1
    ' Row AS INTEGER
    ' Col AS INTEGER
    ' Prob AS SINGLE
    ' Aij AS SINGLE
    X AS STRING * 1
END TYPE
TYPE Fstr2
    round AS INTEGER
    Group AS INTEGER
    SUME AS SINGLE
    SUME2 AS SINGLE
    SUM1 AS SINGLE
    SUM12 AS SINGLE
    SUMX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File1 AS Fstr1, File2 AS Fstr2
CLS
LOCATE 1, 1: PRINT "MMS Negative Skewness Procedure k=2, N = 1200 , M = 480 "
PRINT "Calculating....."
'
'***** Negative Skewness Distribution *****
N& = 1200: M& = 480
'
'k=6 by 1,000x for the test Negative Skewness Distribution
'
LOCATE 1, 1: PRINT "MMS Negative skewness Procedure k=6, N = 1200 , M = 480 "
ns& = 200: ms& = 80
OPEN "R", #1, "MMS200-2.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 12
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 6) / 6 + 1
RR% = 1000
FOR round% = 1 TO RR%
    LOCATE 1, 1: PRINT round%
    ' Group-1 n=200
    REDIM Check1%(1 TO N&)
    FOR I% = 1 TO ns&
        DO
            A& = RandInt&(1, N&)
            LOOP UNTIL Check1%(A&) = 0
            Check1%(A&) = 1
        NEXT I%

```

```

' Group-2 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 2
  NEXT I%
' Group-3 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 3
  NEXT I%
' Group-4 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 4
  NEXT I%
' Group-5 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 5
  NEXT I%
' Group-6 n=200
FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 6
  END IF
NEXT I%
REDIM Check2%(1 TO M&)

' Group-1 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3

```

```

NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 4
  NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 5
  NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 6
  END IF
NEXT I%
NEXT round%
FOR round% = RR% + 1 TO 1000
  LOCATE 3, 1: PRINT "N = 200: "; round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

' Group-1 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 1
  NEXT I%
' Group-2 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 2
  NEXT I%
' Group-3 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 3
  NEXT I%
' Group-4 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 4
  NEXT I%
' Group-5 n=200

```

```

FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 5
  NEXT I%
' Group-6 n=200
FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 6
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 4
  NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 5
  NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 6
  END IF
NEXT I%

```

```

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0: SumX5Sqr! = 0: SumX6Sqr! = 0
OPEN "R", #2, "X1-2.DAT", LEN(File1)
FOR I% = 1 TO N&
    LOCATE 3, 70: PRINT I%
    RecordStart& = (I% - 1) * M& + 1
    FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
        SELECT CASE Check1%(I%)
            CASE 1
                IF Check2%(Record& - RecordStart& + 1) = 1 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 2
                IF Check2%(Record& - RecordStart& + 1) = 2 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 3
                IF Check2%(Record& - RecordStart& + 1) = 3 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 4
                IF Check2%(Record& - RecordStart& + 1) = 4 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 5
                IF Check2%(Record& - RecordStart& + 1) = 5 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX5Sqr! = SumX5Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE ELSE
                IF Check2%(Record& - RecordStart& + 1) = 6 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX6Sqr! = SumX6Sqr! + (VAL(File1.X) ^ 2)
                END IF
            END SELECT
        NEXT Record&
    NEXT I%
CLOSE #2
***** Caluate Stat *****

```

```

SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
SE5! = 0: SE5Sqr! = 0: SI5! = 0: SI5Sqr! = 0
SE6! = 0: SE6Sqr! = 0: SI6! = 0: SI6Sqr! = 0

```

```
FOR I% = 1 TO N&
```

```
  SELECT CASE Check1%(I%)
```

```
    CASE 1
```

```
      SE1! = SE1! + EScore!(I%)
```

```
      SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
```

```
    CASE 2
```

```
      SE2! = SE2! + EScore!(I%)
```

```
      SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
```

```
    CASE 3
```

```
      SE3! = SE3! + EScore!(I%)
```

```
      SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
```

```
    CASE 4
```

```
      SE4! = SE4! + EScore!(I%)
```

```
      SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
```

```
    CASE 5
```

```
      SE5! = SE5! + EScore!(I%)
```

```
      SE5Sqr! = SE5Sqr! + (EScore!(I%) ^ 2)
```

```
    CASE ELSE
```

```
      SE6! = SE6! + EScore!(I%)
```

```
      SE6Sqr! = SE6Sqr! + (EScore!(I%) ^ 2)
```

```
  END SELECT
```

```
NEXT I%
```

```
FOR J% = 1 TO M&
```

```
  SELECT CASE Check2%(J%)
```

```
    CASE 1
```

```
      SI1! = SI1! + IScore!(J%)
```

```
      SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
```

```
    CASE 2
```

```
      SI2! = SI2! + IScore!(J%)
```

```
      SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
```

```
    CASE 3
```

```
      SI3! = SI3! + IScore!(J%)
```

```
      SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
```

```
    CASE 4
```

```
      SI4! = SI4! + IScore!(J%)
```

```
      SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
```

```
    CASE 5
```

```
      SI5! = SI5! + IScore!(J%)
```

```
      SI5Sqr! = SI5Sqr! + (IScore!(J%) ^ 2)
```

```
    CASE ELSE
```

```
      SI6! = SI6! + IScore!(J%)
```

```
      SI6Sqr! = SI6Sqr! + (IScore!(J%) ^ 2)
```

```
  END SELECT
```

```
NEXT J%
```

```
File2.round = round%
```

```
File2.Group = 1
```

```
File2.SUME = SE1!
```

```
File2.SUME2 = SE1Sqr!
```

```
File2.SUMI = SI1!
```

```

File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (round% - 1) * 6 + 1, File2

File2.round = round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (round% - 1) * 6 + 2, File2

File2.round = round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (round% - 1) * 6 + 3, File2

File2.round = round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (round% - 1) * 6 + 4, File2

File2.round = round%
File2.Group = 5
File2.SUME = SE5!
File2.SUME2 = SE5Sqr!
File2.SUMI = SI5!
File2.SUMI2 = SI5Sqr!
File2.SUMX2 = SumX5Sqr!
PUT #1, (round% - 1) * 6 + 5, File2

File2.round = round%
File2.Group = 6
File2.SUME = SE6!
File2.SUME2 = SE6Sqr!
File2.SUMI = SI6!
File2.SUMI2 = SI6Sqr!
File2.SUMX2 = SumX6Sqr!
PUT #1, (round% - 1) * 6 + 6, File2
CLOSE #1
OPEN "R", #1, "MMS200-2.DAT", LEN(File2)
NEXT round%
CLOSE #1

```

'k=4 by 1,000x for the test Negative Skewness Distribution

LOCATE 1, 1: PRINT "MMS Negative skewness Procedure k=4, N = 1200 , M = 480 "

ns& = 300: ms& = 120

OPEN "R", #1, "MMS300-2.DAT", LEN(File2)

REDIM Check1%(1 TO N&)

REDIM Check2%(1 TO M&)

REDIM EScore!(1 TO N&)

REDIM IScore!(1 TO M&)

RANDOMIZE 11

Rec% = LOF(1) \ LEN(File2)

RR% = (Rec% - 4) / 4 + 1

RR% = 787

FOR round% = 1 TO RR%

LOCATE 1, 1: PRINT round%

REDIM Check1%(1 TO N&)

' Group-1 n=300

FOR I% = 1 TO ns&

DO

A& = RandInt&(1, N&)

LOOP UNTIL Check1%(A&) = 0

Check1%(A&) = 1

NEXT I%

' Group-2 n=300

FOR I% = 1 TO ns&

DO .

A& = RandInt&(1, N&)

LOOP UNTIL Check1%(A&) = 0

Check1%(A&) = 2

NEXT I%

' Group-3 n=300

FOR I% = 1 TO ns&

DO

A& = RandInt&(1, N&)

LOOP UNTIL Check1%(A&) = 0

Check1%(A&) = 3

NEXT I%

' Group-4 n=300

FOR I% = 1 TO N&

IF Check1%(I%) = 0 THEN

Check1%(I%) = 4

END IF

NEXT I%

***** Sampling Examinees *****

REDIM Check2%(1 TO M&)

' Group-1 m=120

FOR I% = 1 TO ms&

DO

A& = RandInt&(1, M&)

LOOP UNTIL Check2%(A&) = 0

Check2%(A&) = 1

NEXT I%

' Group-2 m=120

FOR I% = 1 TO ms&

DO

```

        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 2
    NEXT I%
' Group-3 m=120
    FOR I% = 1 TO ms&
        DO
            A& = RandInt&(1, M&)
            LOOP UNTIL Check2%(A&) = 0
            Check2%(A&) = 3
        NEXT I%
' Group-4 m=120
    FOR I% = 1 TO M&
        IF Check2%(I%) = 0 THEN
            Check2%(I%) = 4
        END IF
    NEXT I%
NEXT round%

```

```

FOR round% = RR% + 1 TO 1000
    LOCATE 3, 1: PRINT "N = 300: "; round%
    ***** Sampling Item *****
    REDIM Check1%(1 TO N&)

' Group-1 n=300
    FOR I% = 1 TO ns&
        DO
            A& = RandInt&(1, N&)
            LOOP UNTIL Check1%(A&) = 0
            Check1%(A&) = 1
        NEXT I%
' Group-2 n=300
    FOR I% = 1 TO ns&
        DO
            A& = RandInt&(1, N&)
            LOOP UNTIL Check1%(A&) = 0
            Check1%(A&) = 2
        NEXT I%
' Group-3 n=300
    FOR I% = 1 TO ns&
        DO
            A& = RandInt&(1, N&)
            LOOP UNTIL Check1%(A&) = 0
            Check1%(A&) = 3
        NEXT I%
' Group-4 n=300
    FOR I% = 1 TO N&
        IF Check1%(I%) = 0 THEN
            Check1%(I%) = 4
        END IF
    NEXT I%
    ***** Sampling Examinees *****
    REDIM Check2%(1 TO M&)

' Group-1 m=120

```

```

FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=120
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 4
  END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0
'OPEN "R", #2, "P1-2.DAT", LEN(File1)
OPEN "R", #2, "X1-2.DAT", LEN(File1)
FOR I% = 1 TO N&
  LOCATE 3, 70: PRINT I%
  RecordStart& = (I% - 1) * M& + 1
  FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    LOCATE 2, 70: PRINT (Record& - RecordStart& + 1); ", "; Check2%(Record& - RecordStart& + 1)
    SELECT CASE Check1%(I%)
      CASE 1
        IF Check2%(Record& - RecordStart& + 1) = 1 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
        END IF
      CASE 2
        IF Check2%(Record& - RecordStart& + 1) = 2 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
        END IF
      CASE 3
        IF Check2%(Record& - RecordStart& + 1) = 3 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)

```

```

        SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
    END IF
CASE ELSE
    IF Check2%(Record& - RecordStart& + 1) = 4 THEN
        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
    END IF
END SELECT
NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
FOR I% = 1 TO N&
    SELECT CASE Check1%(I%)
    CASE 1
        SE1! = SE1! + EScore!(I%)
        SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE 2
        SE2! = SE2! + EScore!(I%)
        SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    CASE 3
        SE3! = SE3! + EScore!(I%)
        SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
        SE4! = SE4! + EScore!(I%)
        SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
    END SELECT
NEXT I%
FOR J% = 1 TO M&
    SELECT CASE Check2%(J%)
    CASE 1
        SI1! = SI1! + IScore!(J%)
        SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
    CASE 2
        SI2! = SI2! + IScore!(J%)
        SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
    CASE 3
        SI3! = SI3! + IScore!(J%)
        SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
    CASE ELSE
        SI4! = SI4! + IScore!(J%)
        SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
    END SELECT
NEXT J%

File2.round = round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!

```

```

File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (round% - 1) * 4 + 1, File2

File2.round = round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (round% - 1) * 4 + 2, File2

File2.round = round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (round% - 1) * 4 + 3, File2

File2.round = round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (round% - 1) * 4 + 4, File2
CLOSE #1
OPEN "R", #1, "MMS300-2.DAT", LEN(File2)
NEXT round%
CLOSE #1

'
***** Negative Skewness Distribution *****
N& = 1200: M& = 480
'
' k=2 by 1,000x for the test Neggative Skewness Distribution
'
ns& = 600: ms& = 240
OPEN "R", #1, "MMS600-2.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 10
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 2) / 2 + 1
FOR round% = 1 TO RR%

***** Sampling Item *****
REDIM Check1%(1 TO N&)

' Group-1 n=600

```

```

FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 1
  NEXT I%
' Group-2 n=600
FOR I% = 1 TO N&
  IF Check1%(I%) <> 1 THEN
    Check1%(I%) = 2
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=240
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=240
FOR I% = 1 TO M&
  IF Check2%(I%) <> 1 THEN
    Check2%(I%) = 2
  END IF
NEXT I%
NEXT round%
FOR round% = RR% + 1 TO 1000
  LOCATE 3, 1: PRINT "N = 600: "; round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  ' Group-1 n=600
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=600
  FOR I% = 1 TO N&
    IF Check1%(I%) <> 1 THEN
      Check1%(I%) = 2
    END IF
  NEXT I%
  ***** Sampling Examinees *****
  REDIM Check2%(1 TO M&)

  ' Group-1 m=240
  FOR I% = 1 TO ms&
    DO
      A& = RandInt&(1, M&)
      LOOP UNTIL Check2%(A&) = 0
      Check2%(A&) = 1
    NEXT I%
  ' Group-2 m=240
  FOR I% = 1 TO M&
    IF Check2%(I%) <> 1 THEN
      Check2%(I%) = 2
    END IF
  NEXT I%
NEXT round%

```

```

NEXT I%
' Group-2 m=240
FOR I% = 1 TO M&
  IF Check2%(I%) <> 1 THEN
    Check2%(I%) = 2
  END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0
'OPEN "R", #2, "P1-2.DAT", LEN(File1)
OPEN "R", #2, "X1-2.DAT", LEN(File1)
FOR I% = 1 TO N&
  LOCATE 3, 70: PRINT I%
  RecordStart& = (I% - 1) * M& + 1
  FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    SELECT CASE Check1%(I%)
      CASE 1
        IF Check2%(Record& - RecordStart& + 1) = 1 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
        END IF
      CASE ELSE
        IF Check2%(Record& - RecordStart& + 1) = 2 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
        END IF
    END SELECT
  NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
FOR I% = 1 TO N&
  SELECT CASE Check1%(I%)
    CASE 1
      SE1! = SE1! + EScore!(I%)
      SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
      SE2! = SE2! + EScore!(I%)
      SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    END SELECT
NEXT I%
FOR J% = 1 TO M&
  SELECT CASE Check2%(J%)
    CASE 1
      SI1! = SI1! + IScore!(J%)
      SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
    CASE ELSE
      SI2! = SI2! + IScore!(J%)

```

```

    SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
END SELECT
NEXT J%

File2.round = round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
'A! = (SE1Sqr! / ms& - ((SE1!) ^ 2 / (ns& * ms&))) / (ns& - 1)
'B! = (SI1Sqr! / ns& - ((SI1!) ^ 2 / (ns& * ms&))) / (ms& - 1)
'C! = (SumX1Sqr! - SE1Sqr! / ms& - SI1Sqr! / ns& + ((SE1! ^ 2) / (ns& * ms&))) / ((ns& - 1) * (ms& - 1))
'Variance! = ((N& - 1) / N&) * (A! - (1 - (ms& / M&)) * C!) / ms&
File2.SUMX2 = SumX1Sqr!
PUT #1, (round% - 1) * 2 + 1, File2
File2.round = round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (round% - 1) * 2 + 2, File2
CLOSE #1
OPEN "R", #1, "MMS600-2.DAT", LEN(File2)
NEXT round%
CLOSE #1
END

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())

IF Low& < High& THEN

' Only two elements in this subdivision; swap them if they are out of
' order, then end recursive calls:
IF High& - Low& = 1 THEN
    IF SortArray!(Low&) > SortArray!(High&) THEN
        SWAP SortArray!(Low&), SortArray!(High&)
    END IF
ELSE

' Pick a pivot element at random, then move it to the end:
RandIndex = RandInt&(Low&, High&)
SWAP SortArray!(High&), SortArray!(RandIndex)
Partition! = SortArray!(High&)
DO

' Move in from both sides towards the pivot element:
I& = Low&: J& = High&
DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
    I& = I& + 1
LOOP
DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
    J& = J& - 1

```

```

LOOP

' If we haven't reached the pivot element, it means that two
' elements on either side are out of order, so swap them:
IF I& < J& THEN
    SWAP SortArray!(I&), SortArray!(J&)
END IF
LOOP WHILE I& < J&

' Move the pivot element back to its proper place in the array:
SWAP SortArray(I&), SortArray(High&)

' Recursively call the QuickSort procedure (pass the smaller
' subdivision first to use less stack space):
IF (I& - Low&) < (High& - I&) THEN
    QuickSort Low&, I& - 1, SortArray!()
    QuickSort I& + 1, High&, SortArray!()
ELSE
    QuickSort I& + 1, High&, SortArray!()
    QuickSort Low&, I& - 1, SortArray!()
END IF
END IF
END IF

END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
    RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

```

โปรแกรม AB3-3.BAS ใช้สำหรับกระบวนการสุ่มแบบเมตริกซ์เชิงพหุคูณ กรณีข้อสอบมีลักษณะการแจกแจงค่าความยาก
ของข้อสอบแบบเบ้ขวา

AB3-3.BAS

DECLARE FUNCTION RandInt& (lower&, Upper&)

***** MMS PROCESS *****

' \$DYNAMIC

CLS

'

TYPE Fstr

Mark AS SINGLE

END TYPE

TYPE Fstr1

X AS STRING * 1

END TYPE

TYPE Fstr2

Round AS INTEGER

Group AS INTEGER

SUME AS SINGLE

SUME2 AS SINGLE

SUMI AS SINGLE

SUMI2 AS SINGLE

SUMX2 AS SINGLE

END TYPE

DIM SHARED File AS Fstr, File1 AS Fstr1, File2 AS Fstr2

CLS

PRINT "Calculating....."

'

***** Normal Distribution *****

N& = 1200: M& = 480

'

' k=2 by 1,000x for the test Normal Distribution

'

ns& = 600: ms& = 240

OPEN "R", #1, "MMS600-3.DAT", LEN(File2)

DIM Check1%(1 TO N&)

DIM Check2%(1 TO M&)

DIM EScore!(1 TO N&)

DIM IScore!(1 TO M&)

RANDOMIZE 10

Rcc% = LOF(1) \ LEN(File2)

RR% = (Rcc% - 2) / 2 + 1

FOR Round% = 1 TO RR%

***** Sampling Item *****

REDIM Check1%(1 TO N&)

' Group-1 n=600

FOR I% = 1 TO ns&

DO

A& = RandInt&(1, N&)

LOOP UNTIL Check1%(A&) = 0

Check1%(A&) = 1

NEXT I%

' Group-2 n=600

FOR I% = 1 TO N&

IF Check1%(I%) <> 1 THEN

Check1%(I%) = 2

END IF

NEXT I%

```

***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=240
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=240
FOR I% = 1 TO M&
  IF Check2%(I%) <> 1 THEN
    Check2%(I%) = 2
  END IF
NEXT I%
NEXT Round%
FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 600: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  ' Group-1 n=600
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=600
  FOR I% = 1 TO N&
    IF Check1%(I%) <> 1 THEN
      Check1%(I%) = 2
    END IF
  NEXT I%
  ***** Sampling Examinees *****
  REDIM Check2%(1 TO M&)

  ' Group-1 m=240
  FOR I% = 1 TO ms&
    DO
      A& = RandInt&(1, M&)
      LOOP UNTIL Check2%(A&) = 0
      Check2%(A&) = 1
    NEXT I%
  ' Group-2 m=240
  FOR I% = 1 TO M&
    IF Check2%(I%) <> 1 THEN
      Check2%(I%) = 2
    END IF
  NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0
OPEN "R", #2, "X1-3.DAT", LEN(File1)
FOR I% = 1 TO N&

```

```

LOCATE 1, 70: PRINT I%
RecordStart& = (I% - 1) * M& + 1
FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    SELECT CASE Check1%(I%)
    CASE 1
        IF Check2%(Record& - RecordStart& + 1) = 1 THEN
            GET #2, Record&, File1
            EScore!(I%) = EScore!(I%) + VAL(File1.X)
            IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
            SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
        END IF
    CASE ELSE
        IF Check2%(Record& - RecordStart& + 1) = 2 THEN
            GET #2, Record&, File1
            EScore!(I%) = EScore!(I%) + VAL(File1.X)
            IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
            SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
        END IF
    END SELECT
NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
FOR I% = 1 TO N&
    SELECT CASE Check1%(I%)
    CASE 1
        SE1! = SE1! + EScore!(I%)
        SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
        SE2! = SE2! + EScore!(I%)
        SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    END SELECT
NEXT I%
FOR J% = 1 TO M&
    SELECT CASE Check2%(J%)
    CASE 1
        SI1! = SI1! + IScore!(J%)
        SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
    CASE ELSE
        SI2! = SI2! + IScore!(J%)
        SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
    END SELECT
NEXT J%

File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 2 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!

```

```

File2.SUME2 = SE2Sqr!
File2.SUM1 = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 2 + 2, File2
CLOSE #1
OPEN "R", #1, "MMS600-3.DAT", LEN(File2)
NEXT Round%
CLOSE #1

,
***** Normal Distribution *****
N& = 1200: M& = 480
,
' k=4 by 1,000x for the test Normal Distribution
,
ns& = 300: ms& = 120
OPEN "R", #1, "MMS300-3.DAT", LEN(File2)
REDIM Check1%(1 TO N&)
REDIM Check2%(1 TO M&)
REDIM EScore!(1 TO N&)
REDIM IScore!(1 TO M&)
RANDOMIZE 11
Rec% = LOF(1) \ LEN(File2)
RR% = (Rec% - 4) / 4 + 1

FOR Round% = 1 TO RR%
  REDIM Check1%(1 TO N&)
  ' Group-1 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=300
  FOR I% = 1 TO N&
    IF Check1%(I%) = 0 THEN
      Check1%(I%) = 4
    END IF
  NEXT I%
  ***** Sampling Examinees *****
  REDIM Check2%(1 TO M&)

```

```

' Group-1 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=120
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 4
  END IF
NEXT I%
NEXT Round%

FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 300: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)

  ' Group-1 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=300
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=300

```

```

FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 4
  END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=120
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=120
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 4
  END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0
OPEN "R", #2, "X1-3.DAT", LEN(File1)
FOR I% = 1 TO N&
  LOCATE 1, 70: PRINT I%
  RecordStart& = (I% - 1) * M& + 1
  FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
    LOCATE 2, 70: PRINT (Record& - RecordStart& + 1); ", "; Check2%(Record& - RecordStart& + 1)
    SELECT CASE Check1%(I%)
      CASE 1
        IF Check2%(Record& - RecordStart& + 1) = 1 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)
          IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
          SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
        END IF
      CASE 2
        IF Check2%(Record& - RecordStart& + 1) = 2 THEN
          GET #2, Record&, File1
          EScore!(I%) = EScore!(I%) + VAL(File1.X)

```

```

        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
    END IF
CASE 3
    IF Check2%(Record& - RecordStart& + 1) = 3 THEN
        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
    END IF
CASE ELSE
    IF Check2%(Record& - RecordStart& + 1) = 4 THEN
        GET #2, Record&, File1
        EScore!(I%) = EScore!(I%) + VAL(File1.X)
        IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
        SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
    END IF
END SELECT
NEXT Record&
NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
FOR I% = 1 TO N&
    SELECT CASE Check1%(I%)
    CASE 1
        SE1! = SE1! + EScore!(I%)
        SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
    CASE 2
        SE2! = SE2! + EScore!(I%)
        SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
    CASE 3
        SE3! = SE3! + EScore!(I%)
        SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
    CASE ELSE
        SE4! = SE4! + EScore!(I%)
        SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
    END SELECT
NEXT I%
FOR J% = 1 TO M&
    SELECT CASE Check2%(J%)
    CASE 1
        SI1! = SI1! + IScore!(J%)
        SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
    CASE 2
        SI2! = SI2! + IScore!(J%)
        SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
    CASE 3
        SI3! = SI3! + IScore!(J%)
        SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
    CASE ELSE
        SI4! = SI4! + IScore!(J%)
        SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
    END SELECT

```

```

NEXT J%
File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!
File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 4 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 4 + 2, File2
File2.Round = Round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (Round% - 1) * 4 + 3, File2
File2.Round = Round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (Round% - 1) * 4 + 4, File2
CLOSE #1
OPEN "R", #1, "MMS300-3.DAT", LEN(File2)
NEXT Round%
CLOSE #1

```

***** Normal Distribution *****

N& = 1200: M& = 480

' k=6 by 1,000x for the test Normal Distribution

ns& = 200: ms& = 80

OPEN "R", #1, "MMS200-3.DAT", LEN(File2)

REDIM Check1%(1 TO N&)

REDIM Check2%(1 TO M&)

REDIM EScore!(1 TO N&)

REDIM IScore!(1 TO M&)

RANDOMIZE 12

Rec% = LOF(1) \ LEN(File2)

RR% = (Rec% - 6) / 6 + 1

FOR Round% = 1 TO RR%

' Group-1 n=200

REDIM Check1%(1 TO N&)

```

FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 1
  NEXT I%
' Group-2 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 2
  NEXT I%
' Group-3 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 3
  NEXT I%
' Group-4 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 4
  NEXT I%
' Group-5 n=200
FOR I% = 1 TO ns&
  DO
    A& = RandInt&(1, N&)
    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 5
  NEXT I%
' Group-6 n=200
FOR I% = 1 TO N&
  IF Check1%(I%) = 0 THEN
    Check1%(I%) = 6
  END IF
NEXT I%
REDIM Check2%(1 TO M&)

' Group-1 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 1
  NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 2
  NEXT I%
' Group-3 m=80

```

```

FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 3
  NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 4
  NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
  DO
    A& = RandInt&(1, M&)
    LOOP UNTIL Check2%(A&) = 0
    Check2%(A&) = 5
  NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
  IF Check2%(I%) = 0 THEN
    Check2%(I%) = 6
  END IF
NEXT I%
NEXT Round%
FOR Round% = RR% + 1 TO 1000
  LOCATE 1, 1: PRINT "N = 200: "; Round%
  ***** Sampling Item *****
  REDIM Check1%(1 TO N&)
  ' Group-1 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 1
    NEXT I%
  ' Group-2 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 2
    NEXT I%
  ' Group-3 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)
      LOOP UNTIL Check1%(A&) = 0
      Check1%(A&) = 3
    NEXT I%
  ' Group-4 n=200
  FOR I% = 1 TO ns&
    DO
      A& = RandInt&(1, N&)

```

```

    LOOP UNTIL Check1%(A&) = 0
    Check1%(A&) = 4
NEXT I%
' Group-5 n=200
FOR I% = 1 TO ns&
    DO
        A& = RandInt&(1, N&)
        LOOP UNTIL Check1%(A&) = 0
        Check1%(A&) = 5
    NEXT I%
' Group-6 n=200
FOR I% = 1 TO N&
    IF Check1%(I%) = 0 THEN
        Check1%(I%) = 6
    END IF
NEXT I%
***** Sampling Examinees *****
REDIM Check2%(1 TO M&)

' Group-1 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 1
    NEXT I%
' Group-2 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 2
    NEXT I%
' Group-3 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 3
    NEXT I%
' Group-4 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 4
    NEXT I%
' Group-5 m=80
FOR I% = 1 TO ms&
    DO
        A& = RandInt&(1, M&)
        LOOP UNTIL Check2%(A&) = 0
        Check2%(A&) = 5
    NEXT I%
' Group-6 m=80
FOR I% = 1 TO M&
    IF Check2%(I%) = 0 THEN

```

```

        Check2%(I%) = 6
    END IF
NEXT I%

'Print Samples & Calculate Item Score & Examinee Score
REDIM EScore!(1 TO N&), IScore!(1 TO M&)
SumX1Sqr! = 0: SumX2Sqr! = 0: SumX3Sqr! = 0: SumX4Sqr! = 0: SumX5Sqr! = 0: SumX6Sqr! = 0
OPEN "R", #2, "X1-3.DAT", LEN(File1)
FOR I% = 1 TO N&
    LOCATE 1, 70: PRINT I%
    RecordStart& = (I% - 1) * M& + 1
    FOR Record& = RecordStart& TO (RecordStart& + M& - 1)
        SELECT CASE Check1%(I%)
            CASE 1
                IF Check2%(Record& - RecordStart& + 1) = 1 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX1Sqr! = SumX1Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 2
                IF Check2%(Record& - RecordStart& + 1) = 2 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX2Sqr! = SumX2Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 3
                IF Check2%(Record& - RecordStart& + 1) = 3 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX3Sqr! = SumX3Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 4
                IF Check2%(Record& - RecordStart& + 1) = 4 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX4Sqr! = SumX4Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE 5
                IF Check2%(Record& - RecordStart& + 1) = 5 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX5Sqr! = SumX5Sqr! + (VAL(File1.X) ^ 2)
                END IF
            CASE ELSE
                IF Check2%(Record& - RecordStart& + 1) = 6 THEN
                    GET #2, Record&, File1
                    EScore!(I%) = EScore!(I%) + VAL(File1.X)
                    IScore!(Record& - RecordStart& + 1) = IScore!(Record& - RecordStart& + 1) + VAL(File1.X)
                    SumX6Sqr! = SumX6Sqr! + (VAL(File1.X) ^ 2)
                END IF
            END SELECT
        NEXT Record&
    NEXT I%

```

```

NEXT I%
CLOSE #2
***** Caluate Stat *****
SE1! = 0: SE1Sqr! = 0: SI1! = 0: SI1Sqr! = 0
SE2! = 0: SE2Sqr! = 0: SI2! = 0: SI2Sqr! = 0
SE3! = 0: SE3Sqr! = 0: SI3! = 0: SI3Sqr! = 0
SE4! = 0: SE4Sqr! = 0: SI4! = 0: SI4Sqr! = 0
SE5! = 0: SE5Sqr! = 0: SI5! = 0: SI5Sqr! = 0
SE6! = 0: SE6Sqr! = 0: SI6! = 0: SI6Sqr! = 0
FOR I% = 1 TO N&
  SELECT CASE Check1%(I%)
  CASE 1
    SE1! = SE1! + EScore!(I%)
    SE1Sqr! = SE1Sqr! + (EScore!(I%) ^ 2)
  CASE 2
    SE2! = SE2! + EScore!(I%)
    SE2Sqr! = SE2Sqr! + (EScore!(I%) ^ 2)
  CASE 3
    SE3! = SE3! + EScore!(I%)
    SE3Sqr! = SE3Sqr! + (EScore!(I%) ^ 2)
  CASE 4
    SE4! = SE4! + EScore!(I%)
    SE4Sqr! = SE4Sqr! + (EScore!(I%) ^ 2)
  CASE 5
    SE5! = SE5! + EScore!(I%)
    SE5Sqr! = SE5Sqr! + (EScore!(I%) ^ 2)
  CASE ELSE
    SE6! = SE6! + EScore!(I%)
    SE6Sqr! = SE6Sqr! + (EScore!(I%) ^ 2)
  END SELECT
NEXT I%
FOR J% = 1 TO M&
  SELECT CASE Check2%(J%)
  CASE 1
    SI1! = SI1! + IScore!(J%)
    SI1Sqr! = SI1Sqr! + (IScore!(J%) ^ 2)
  CASE 2
    SI2! = SI2! + IScore!(J%)
    SI2Sqr! = SI2Sqr! + (IScore!(J%) ^ 2)
  CASE 3
    SI3! = SI3! + IScore!(J%)
    SI3Sqr! = SI3Sqr! + (IScore!(J%) ^ 2)
  CASE 4
    SI4! = SI4! + IScore!(J%)
    SI4Sqr! = SI4Sqr! + (IScore!(J%) ^ 2)
  CASE 5
    SI5! = SI5! + IScore!(J%)
    SI5Sqr! = SI5Sqr! + (IScore!(J%) ^ 2)
  CASE ELSE
    SI6! = SI6! + IScore!(J%)
    SI6Sqr! = SI6Sqr! + (IScore!(J%) ^ 2)
  END SELECT
NEXT J%
File2.Round = Round%
File2.Group = 1
File2.SUME = SE1!
File2.SUME2 = SE1Sqr!

```

```

File2.SUMI = SI1!
File2.SUMI2 = SI1Sqr!
File2.SUMX2 = SumX1Sqr!
PUT #1, (Round% - 1) * 6 + 1, File2
File2.Round = Round%
File2.Group = 2
File2.SUME = SE2!
File2.SUME2 = SE2Sqr!
File2.SUMI = SI2!
File2.SUMI2 = SI2Sqr!
File2.SUMX2 = SumX2Sqr!
PUT #1, (Round% - 1) * 6 + 2, File2
File2.Round = Round%
File2.Group = 3
File2.SUME = SE3!
File2.SUME2 = SE3Sqr!
File2.SUMI = SI3!
File2.SUMI2 = SI3Sqr!
File2.SUMX2 = SumX3Sqr!
PUT #1, (Round% - 1) * 6 + 3, File2
File2.Round = Round%
File2.Group = 4
File2.SUME = SE4!
File2.SUME2 = SE4Sqr!
File2.SUMI = SI4!
File2.SUMI2 = SI4Sqr!
File2.SUMX2 = SumX4Sqr!
PUT #1, (Round% - 1) * 6 + 4, File2
File2.Round = Round%
File2.Group = 5
File2.SUME = SE5!
File2.SUME2 = SE5Sqr!
File2.SUMI = SI5!
File2.SUMI2 = SI5Sqr!
File2.SUMX2 = SumX5Sqr!
PUT #1, (Round% - 1) * 6 + 5, File2
File2.Round = Round%
File2.Group = 6
File2.SUME = SE6!
File2.SUME2 = SE6Sqr!
File2.SUMI = SI6!
File2.SUMI2 = SI6Sqr!
File2.SUMX2 = SumX6Sqr!
PUT #1, (Round% - 1) * 6 + 6, File2
CLOSE #1
OPEN "R", #1, "MMS200-3.DAT", LEN(File2)
NEXT Round%
CLOSE #1
END

```

```

REM $STATIC
SUB QuickSort (Low&, High&, SortArray!())

```

```

IF Low& < High& THEN

```

```

' Only two elements in this subdivision; swap them if they are out of
' order, then end recursive calls:

```

```

IF High& - Low& = 1 THEN
  IF SortArray!(Low&) > SortArray!(High&) THEN
    SWAP SortArray!(Low&), SortArray!(High&)
  END IF
ELSE
  ' Pick a pivot element at random, then move it to the end:
  RandIndex = RandInt&(Low&, High&)
  SWAP SortArray!(High&), SortArray!(RandIndex)
  Partition! = SortArray!(High&)
  DO

    ' Move in from both sides towards the pivot element:
    I& = Low&: J& = High&
    DO WHILE (I& < J&) AND (SortArray!(I&) <= Partition!)
      I& = I& + 1
    LOOP
    DO WHILE (J& > I&) AND (SortArray!(J&) >= Partition!)
      J& = J& - 1
    LOOP

    ' If we haven't reached the pivot element, it means that two
    ' elements on either side are out of order, so swap them:
    IF I& < J& THEN
      SWAP SortArray!(I&), SortArray!(J&)
    END IF
  LOOP WHILE I& < J&

  ' Move the pivot element back to its proper place in the array:
  SWAP SortArray!(I&), SortArray!(High&)

  ' Recursively call the QuickSort procedure (pass the smaller
  ' subdivision first to use less stack space):
  IF (I& - Low&) < (High& - I&) THEN
    QuickSort Low&, I& - 1, SortArray!()
    QuickSort I& + 1, High&, SortArray!()
  ELSE
    QuickSort I& + 1, High&, SortArray!()
    QuickSort Low&, I& - 1, SortArray!()
  END IF
END IF
END IF

END SUB

FUNCTION RandInt& (lower&, Upper&) STATIC
  RandInt& = INT(RND * (Upper& - lower& + 1)) + lower&
END FUNCTION

```

โปรแกรม LSCORE.BAS ใช้คำนวณหาคะแนนของประชากรผู้สอบที่ตอบข้อสอบถูก

'Program Name : LSCORE.BAS

CLS

TYPE Fstr2

Mark AS SINGLE

END TYPE

DIM SHARED File2 AS Fstr2

DIM XX%(60)

***** Normal Distribution *****

OPEN "R", #1, "MP1-1.DAT", LEN(File2)

OPEN "O", #2, "NorO.TXT"

PRINT "Examinee-Normal Item-Normal"

FOR I% = 1 TO LOF(1) \ LEN(File2)

GET #1, I%, File2

PRINT I%, File2.Mark

SELECT CASE (File2.Mark MOD 8)

CASE 0

KK% = INT((File2.Mark / 8))

CASE ELSE

KK% = INT((8 - (File2.Mark MOD 8) + File2.Mark) / 8)

END SELECT

XX%(KK%) = XX%(KK%) + 1

NEXT I%

FOR KK% = 1 TO 60

PRINT #2, USING "#####", XX%(KK%)

NEXT KK%

CLOSE

ERASE XX%

OPEN "R", #1, "MP1-2.DAT", LEN(File2)

OPEN "O", #2, "NegO.TXT"

PRINT "Examinee-Normal Item-Negative Skewness"

FOR I% = 1 TO LOF(1) \ LEN(File2)

GET #1, I%, File2

PRINT I%, File2.Mark

SELECT CASE (File2.Mark MOD 8)

CASE 0

KK% = INT((File2.Mark / 8))

CASE ELSE

KK% = INT((8 - (File2.Mark MOD 8) + File2.Mark) / 8)

END SELECT

XX%(KK%) = XX%(KK%) + 1

NEXT I%

FOR KK% = 1 TO 60

PRINT #2, USING "#####", XX%(KK%)

NEXT KK%

CLOSE

ERASE XX%

OPEN "R", #1, "MP1-3.DAT", LEN(File2)

OPEN "O", #2, "PosO.TXT"

PRINT "Examinee-Normal Item-Positive Skewness"

FOR I% = 1 TO LOF(1) \ LEN(File2)

GET #1, I%, File2

PRINT I%, File2.Mark

SELECT CASE (File2.Mark MOD 8)

CASE 0

โปรแกรม LSRS.BAS ใช้แสดงผลเปรียบเทียบค่าคะแนนเฉลี่ย และความแปรปรวนคะแนนจากการสุ่มแบบง่าย

กับประชากร

```

' LSRS.BAS
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION tprob# (t#, k&)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
DECLARE FUNCTION VVariance! (Var!())
***** SRS PROCESS *****
' $DYNAMIC
TYPE Fstr
  Row AS INTEGER
  Col AS INTEGER
  Prob AS SINGLE
  Uni AS SINGLE
  x AS STRING * 1
END TYPE
TYPE Fstr2
  Round AS INTEGER
  Group AS INTEGER
  SumX AS SINGLE
  SumX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File2 AS Fstr2
DIM SM!(1 TO 1000)
Min! = 0: Max! = 0
CLS
DO
  CLS
  n& = 1200: M& = 480
  DO
    CLS
    PRINT "Result-SRS process to PRINT..."
    PRINT "You want to PRINT (1) Normal (2) Negative skewness (3) Positive skewness ? ";
    Ans1$ = INPUT$(1)
    Ans1$ = UCASE$(Ans1$)
    LOOP UNTIL INSTR(1, "123", Ans1$) > 0
    PRINT Ans1$
    DO
      PRINT "You want to PRINT k = ? ";
      k% = VAL(INPUT$(1))
      LOOP UNTIL k% = 2 OR k% = 4 OR k% = 6
      PRINT k%
      ns& = n& / k%
      PRINT "You want to print on"
      PRINT "(P) Printer, (F) File " + "LSRS" + Ans1$ + LTRIM$(STR$(k%)) + ".OUT," + " (M) Monitor, (S) Stop ?";
      AS = INPUT$(1)
      SELECT CASE UCASE$(AS)
      CASE "P"
        OPEN "O", #2, "LPT1:"
        GOSUB PrintStat
      CASE "F"
        OPEN "O", #2, "LSRS" + Ans1$ + LTRIM$(STR$(k%)) + ".OUT"
        GOSUB PrintStat
      CASE "M"
        OPEN "O", #2, "SCRN:"
        GOSUB PrintStat
      CASE "S"
        CLOSE
        EXIT DO
    
```

```

LSRS-12.BAS
DECLARE FUNCTION Zscore# (Z#)
DECLARE FUNCTION tprob# (t#, k&)
DECLARE FUNCTION SMEAN! (Var!())
DECLARE FUNCTION SVariance! (Var!())
DECLARE FUNCTION VVariance! (Var!())
***** SRS PROCESS *****
'DYNAMIC
TYPE Fstr
    Row AS INTEGER
    Col AS INTEGER
    Prob AS SINGLE
    Uni AS SINGLE
    x AS STRING * 1
END TYPE
TYPE Fstr2
    Round AS INTEGER
    Group AS INTEGER
    SumX AS SINGLE
    SumX2 AS SINGLE
END TYPE
DIM SHARED File AS Fstr, File2 AS Fstr2
DIM SM!(1 TO 1000)
Min! = 0: Max! = 0
CLS
DO
    CLS
    n& = 1200: M& = 480
    DO
        CLS
        PRINT "Result-SRS process to PRINT..."
        PRINT "You want to PRINT (1) Normal (2) Negative skewness (3) Positive skewness ? ";
        Ans1$ = INPUT$(1)
        Ans1$ = UCASE$(Ans1$)
        LOOP UNTIL INSTR(1, "123", Ans1$) > 0
        PRINT Ans1$
        DO
            PRINT "You want to PRINT k = ? ";
            k% = VAL(INPUT$(1))
            LOOP UNTIL k% = 2 OR k% = 4 OR k% = 6
            PRINT k%
            ns& = n& / k%
            PRINT "You want to print on"
            PRINT "(P) Printer, (F) File " + "LSRS" + Ans1$ + LTRIM$(STR$(k%)) + ".OUT," + " (M) Monitor, (S) Stop ?";
            A$ = INPUT$(1)
            SELECT CASE UCASE$(A$)
            CASE "P"
                OPEN "O", #2, "LPT1:"
                GOSUB PrintStat
            CASE "F"
                OPEN "O", #2, "LSRS" + Ans1$ + LTRIM$(STR$(k%)) + ".OUT"
                GOSUB PrintStat
            CASE "M"
                OPEN "O", #2, "SCRN:"
                GOSUB PrintStat
            CASE "S"
                CLOSE
            EXIT DO
        DO
    CLS
    EXIT DO

```

```

CASE ELSE
END SELECT
LOOP
END
PrintStat:
***** PRINT STATISTICS *****
OPEN "T", #1, "MEAN.DAT"
SELECT CASE Ans1$
CASE "1"
  LINE INPUT #1, NN$
  Mn1! = VAL(LEFT$(NN$, 9))
  Var1! = VAL(RIGHT$(NN$, 15))
CASE "2"
  LINE INPUT #1, NN$
  LINE INPUT #1, NN$
  Mn1! = VAL(LEFT$(NN$, 9))
  Var1! = VAL(RIGHT$(NN$, 15))
CASE ELSE
  LINE INPUT #1, NN$
  LINE INPUT #1, NN$
  LINE INPUT #1, NN$
  Mn1! = VAL(LEFT$(NN$, 9))
  Var1! = VAL(RIGHT$(NN$, 15))
END SELECT
CLOSE #1
OPEN "R", #1, "S" + LTRIM$(RTRIM$(STR$(ns&))) + "1-" + Ans1$ + ".DAT", LEN(File2)
GOSUB Head
SumMean! = 0
SumVar! = 0
Max% = 56 * k%
Max% = Max% - (Max% MOD k%)
FOR I% = 1 TO LOF(1) / LEN(File2) STEP Max% * 2
  FOR x% = 1 TO Max%
    GET #1, I% + (x% - 1), File2
    IF File2.Group = 1 THEN
      Mean! = File2.SumX / ns&

      SM!(File2.Round) = Mean!
      IF File2.Round = 1 THEN
        Min! = Mean!: Max! = Mean!
      ELSE
        IF Mean! > Max! THEN
          Max! = Mean!
        ELSE
          IF Mean! < Min! THEN
            Min! = Mean!
          END IF
        END IF
      END IF
    END IF
  END IF

  Variance! = (File2.SumX2 - (File2.SumX ^ 2 / ns&)) / (ns& - 1)
  SumMean! = SumMean! + Mean!
  SumVar! = SumVar! + Variance!
  PRINT #2, TAB(1); USING "###"; File2.Round;
  PRINT #2, TAB(9); USING "#####"; Mean!;
  PRINT #2, TAB(18); USING "###.###.###"; Variance!;
  PRINT #2, TAB(33); USING "###.###.###"; ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)));
  t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))

```

```

tp# = ABS(tprob#(t#, ns& - 1))
IF tp# > .995 THEN
  PRINT #2, "***";
ELSE
  IF tp# > .975 THEN
    PRINT #2, "**";
  END IF
END IF
PRINT #2, TAB(47); USING "###,###.####"; ((ns& - 1) * (Variance! / Var1!));
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
  GET #1, I% + Max% + (x% - 1), File2
  IF File2.Group = 1 THEN
    Mean! = File2.SumX / ns&

    SM!(File2.Round) = Mean!
    IF File2.Round = 1 THEN
      Min! = Mean!: Max! = Mean!
    ELSE
      IF Mean! > Max! THEN
        Max! = Mean!
      ELSE
        IF Mean! < Min! THEN
          Min! = Mean!
        END IF
      END IF
    END IF
  END IF

  Variance! = (File2.SumX2 - (File2.SumX ^ 2 / ns&)) / (ns& - 1)
  SumMean! = SumMean! + Mean!
  SumVar! = SumVar! + Variance!
  PRINT #2, TAB(61); USING "#,###"; File2.Round;
  PRINT #2, TAB(69); USING "###.####"; Mean!;
  PRINT #2, TAB(78); USING "#,###.#### "; Variance!;
  t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))
  PRINT #2, USING "#,###.####"; t#;
  tp# = ABS(tprob#(t#, ns& - 1))
  IF tp# > .995 THEN
    PRINT #2, "***";
  ELSE
    IF tp# > .975 THEN
      PRINT #2, "**";
    ELSE
      PRINT #2, " ";
    END IF
  END IF
  PRINT #2, USING "###,###.####"; ((ns& - 1) * (Variance! / Var1!))
END IF
END IF
END IF
NEXT x%
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
  PRINT #2, CHR$(12)
  GOSUB Head
END IF
NEXT I%
PRINT #2, TAB(1); "-----"
Mean! = SumMean! / 1000
Variance! = SumVar! / 1000

```

```

PRINT #2, TAB(1); "TOTAL";
PRINT #2, TAB(9); USING "####.####"; Mean!;
PRINT #2, TAB(18); USING "###.###.####"; Variance!;
PRINT #2, TAB(33); USING "###.####"; ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)));
t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))
tp# = ABS(ABS(tprob#(t#, ns& - 1)))
IF tp# > .995 THEN
    PRINT #2, "****";
ELSE
    IF tp# > .975 THEN
        PRINT #2, "***";
    END IF
END IF
PRINT #2, TAB(47); USING "###.###.####"; ((ns& - 1) * (Variance! / Var1!))
PRINT #2, "-----"
PRINT #2, USING "Min mean = ####.####, "; Min!;
PRINT #2, USING "Max mean = ####.####, "; Max!;
PRINT #2, USING "Variance of means = ###.####"; VVariance!(SM!())
CLOSE #1
SELECT CASE UCASE$(A$)
CASE "M"
CASE ELSE
    PRINT #2, CHR$(12)
END SELECT
END
Head:
WIDTH #2, 255
IF Ans1$ = "1" THEN
    PRINT #2, "SRS PROCESS : in Normal Distribution :-"
ELSEIF Ans1$ = "2" THEN
    PRINT #2, "SRS PROCESS : in Negative skewness Distribution :-"
ELSE
    PRINT #2, "SRS PROCESS : in Positive skewness Distribution :-"
END IF
PRINT #2, USING "when k = #, "; k%;
PRINT #2, USING "n = ###, "; ns&;
PRINT #2, USING "with mean = ####.#### "; Mn1!;
PRINT #2, USING "variance = ###.###.#### "; Var1!
PRINT #2, "-----"
PRINT #2, "No. Mean Variance t-value Chi-Square No. Mean Variance t-value Chi-Square"
PRINT #2, "-----"
RETURN
END

REM $STATIC
FUNCTION SMEAN! (Var!())
    M! = 0
    FOR Number& = 1 TO UBOUND(Var!)
        M! = M! + Var!(Number&)
    NEXT Number&
    SMEAN! = M! / UBOUND(Var!)
END FUNCTION

FUNCTION SVariance! (Var!())
    M! = SMEAN!(Var!())
    S! = 0

```

```

FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - M!) ^ 2
NEXT Number&
SVariance! = S! / (UBOUND(Var!) - 1)
END FUNCTION

FUNCTION tprob# (t#, k&)
'parameter tprob = probability when t<=t#
'    t# = t-score value to find prob.
'    k& = degree of freedom
'
'if k&>30 then limit of t-score to Z-score
IF k& > 30 THEN
    tprob# = Zscore#(t#)
    EXIT FUNCTION
END IF
'This function calculated by seeing Abramowitz & Stegun (1972)
Zeta# = ATN(t# / SQR(k&))
SELECT CASE k&
CASE 1
    A# = 2 * Zeta# / pi
CASE ELSE
    odd& = (k& MOD 2) = 1)
    SELECT CASE odd&
    CASE -1 'odd
        IF k& = 3 THEN
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * COS(Zeta#))
        ELSE
            CosZeta# = COS(Zeta#)
            FOR x& = 2 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 2 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * CosZeta#)
        END IF
    CASE ELSE 'even
        IF k& = 2 THEN
            A# = SIN(Zeta#)
        ELSE
            CosZeta# = 1
            FOR x& = 1 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 1 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = SIN(Zeta#) * CosZeta#
        END IF
    END SELECT
END SELECT
tprob# = .5 * (1 + A#)
END FUNCTION

```

```
FUNCTION VVariance! (Var!())
```

```
  M! = SMEAN!(Var!())
```

```
  S! = 0
```

```
  FOR Number& = 1 TO UBOUND(Var!)
```

```
    S! = S! + (Var!(Number&) - M!) ^ 2
```

```
  NEXT Number&
```

```
  VVariance! = S! / UBOUND(Var!)
```

```
END FUNCTION
```

```
FUNCTION Zscore# (Z#)
```

```
' The formula used from Abramowitz & Stegun (1972) formula 2
```

```
IF Z# >= -14.14 AND Z# <= 14.14 THEN
```

```
  Y# = (.7071067812#) * Z#
```

```
ELSE
```

```
  Y# = 10
```

```
END IF
```

```
A1# = 7.052307839999999D-02
```

```
A2# = .0422820123#
```

```
A3# = .0092705272#
```

```
A4# = .0001520143#
```

```
A5# = .0002765672#
```

```
A6# = .0000430638#
```

```
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) + A6# * (Y# ^ 6)) ^ (-16)
```

```
END FUNCTION
```

```

CASE ELSE
END SELECT
LOOP
END
PrintStat:
***** PRINT STATISTICS *****
OPEN "I", #1, "MEAN.DAT"
SELECT CASE Ans1$
CASE "1"
    LINE INPUT #1, NN$
    Mn1! = VAL(LEFT$(NN$, 9))
    Var1! = VAL(RIGHT$(NN$, 15))
CASE "2"
    LINE INPUT #1, NN$
    LINE INPUT #1, NN$
    Mn1! = VAL(LEFT$(NN$, 9))
    Var1! = VAL(RIGHT$(NN$, 15))
CASE ELSE
    LINE INPUT #1, NN$
    LINE INPUT #1, NN$
    LINE INPUT #1, NN$
    Mn1! = VAL(LEFT$(NN$, 9))
    Var1! = VAL(RIGHT$(NN$, 15))
END SELECT
CLOSE #1
OPEN "R", #1, "S" + LTRIM$(RTRIM$(STR$(ns&))) + "1-" + Ans1$ + ".DAT", LEN(File2)
GOSUB Head
SumMean! = 0
SumVar! = 0
Max% = 56 * k%
Max% = Max% - (Max% MOD k%)
FOR I% = 1 TO LOF(1) / LEN(File2) STEP Max% * 2
    FOR x% = 1 TO Max%
        GET #1, I% + (x% - 1), File2
        IF File2.Group = 1 THEN
            Mean! = File2.SumX / ns&

            SM!(File2.Round) = Mean!
            IF File2.Round = 1 THEN
                Min! = Mean!: Max! = Mean!
            ELSE
                IF Mean! > Max! THEN
                    Max! = Mean!
                ELSE
                    IF Mean! < Min! THEN
                        Min! = Mean!
                    END IF
                END IF
            END IF

            Variance! = (File2.SumX2 - (File2.SumX ^ 2 / ns&)) / (ns& - 1)
            SumMean! = SumMean! + Mean!
            SumVar! = SumVar! + Variance!
            PRINT #2, TAB(1); USING "###"; File2.Round;
            PRINT #2, TAB(9); USING "####.####"; Mean!;
            PRINT #2, TAB(18); USING "###.###.###"; Variance!;
            PRINT #2, TAB(33); USING "###.####"; ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)));
            t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))

```

```

tp# = ABS(tprob#(t#, ns& - 1))
IF tp# > .995 THEN
    PRINT #2, "****";
ELSE
    IF tp# > .975 THEN
        PRINT #2, "***";
    END IF
END IF
PRINT #2, TAB(47); USING "###,###.####"; ((ns& - 1) * (Variance! / Var1!));
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
    GET #1, I% + Max% + (x% - 1), File2
    IF File2.Group = 1 THEN
        Mean! = File2.SumX / ns&

        SM!(File2.Round) = Mean!
        IF File2.Round = 1 THEN
            Min! = Mean!: Max! = Mean!
        ELSE
            IF Mean! > Max! THEN
                Max! = Mean!
            ELSE
                IF Mean! < Min! THEN
                    Min! = Mean!
                END IF
            END IF
        END IF
    END IF
    Variance! = (File2.SumX2 - (File2.SumX ^ 2 / ns&)) / (ns& - 1)
    SumMean! = SumMean! + Mean!
    SumVar! = SumVar! + Variance!
    PRINT #2, TAB(61); USING "#,###"; File2.Round;
    PRINT #2, TAB(69); USING "###.####"; Mean!;
    PRINT #2, TAB(78); USING "#,###.####"; Variance!;
    t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))
    PRINT #2, USING "###.####"; t#;
    tp# = ABS(tprob#(t#, ns& - 1))
    IF tp# > .995 THEN
        PRINT #2, "****";
    ELSE
        IF tp# > .975 THEN
            PRINT #2, "***";
        ELSE
            PRINT #2, " ";
        END IF
    END IF
    PRINT #2, USING "###,###.####"; ((ns& - 1) * (Variance! / Var1!))
END IF
END IF
END IF
NEXT x%
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
    PRINT #2, CHR$(12)
    GOSUB Head
END IF
NEXT I%
PRINT #2, TAB(1); "-----"
Mean! = SumMean! / 1000
Variance! = SumVar! / 1000

```

```

PRINT #2, TAB(1); "TOTAL";
PRINT #2, TAB(9); USING "####.####"; Mean!;
PRINT #2, TAB(18); USING "###.###.####"; Variance!;
PRINT #2, TAB(33); USING "###.###.####"; ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)));
t# = ((Mean! - Mn1!) / (SQR(Var1!) / SQR(ns&)))
tp# = ABS(ABS(tprob#(t#, ns& - 1)))
IF tp# > .995 THEN
    PRINT #2, "****";
ELSE
    IF tp# > .975 THEN
        PRINT #2, "***";
    END IF
END IF
PRINT #2, TAB(47); USING "###.###.####"; ((ns& - 1) * (Variance! / Var1!))
PRINT #2, "-----"
PRINT #2, USING "Min mean = ####.####, "; Min1;
PRINT #2, USING "Max mean = ####.####, "; Max1;
PRINT #2, USING "Variance of means = ###.###.####"; VVariance!(SM1())
CLOSE #1
SELECT CASE UCASE$(A$)
CASE "M"
CASE ELSE
    PRINT #2, CHR$(12)
END SELECT
END
Head:
WIDTH #2, 255
IF Ans1$ = "1" THEN
    PRINT #2, "SRS PROCESS : in Normal Distribution :-"
ELSEIF Ans1$ = "2" THEN
    PRINT #2, "SRS PROCESS : in Negative skewness Distribution :-"
ELSE
    PRINT #2, "SRS PROCESS : in Positive skewness Distribution :-"
END IF
PRINT #2, USING "when k = #, "; k%;
PRINT #2, USING "n = ###, "; ns&;
PRINT #2, USING "with mean = ###.#### "; Mn1!;
PRINT #2, USING "variance = ###.###.#### "; Var1!
PRINT #2, "-----"
PRINT #2, " No. Mean Variance t-value Chi-Square No. Mean Variance t-value Chi-Square"
PRINT #2, "-----"
RETURN
END

REM $STATIC
FUNCTION SMEAN! (Var1())
    M! = 0
    FOR Number& = 1 TO UBOUND(Var1)
        M! = M! + Var1(Number&)
    NEXT Number&
    SMEAN! = M! / UBOUND(Var1)
END FUNCTION

FUNCTION SVariance! (Var1())
    M! = SMEAN!(Var1())
    S! = 0

```

```

FOR Number& = 1 TO UBOUND(Var!)
    S! = S! + (Var!(Number&) - M!) ^ 2
NEXT Number&
SVariance! = S! / (UBOUND(Var!) - 1)
END FUNCTION

```

```

FUNCTION tprob# (t#, k&)
'parameter tprob = probability when t<=t#
'    t# = t-score value to find prob.
'    k& = degree of freedom
'
'if k&>30 then limit of t-score to Z-score
IF k& > 30 THEN
    tprob# = Zscore#(t#)
    EXIT FUNCTION
END IF
' This function calculated by seeing Abramowitz & Stegun (1972)
Zeta# = ATN(t# / SQR(k&))
SELECT CASE k&
CASE 1
    A# = 2 * Zeta# / pi
CASE ELSE
    odd& = ((k& MOD 2) = 1)
    SELECT CASE odd&
    CASE -1 'odd
        IF k& = 3 THEN
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * COS(Zeta#))
        ELSE
            CosZeta# = COS(Zeta#)
            FOR x& = 2 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 2 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * CosZeta#)
        END IF
    CASE ELSE 'even
        IF k& = 2 THEN
            A# = SIN(Zeta#)
        ELSE
            CosZeta# = 1
            FOR x& = 1 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 1 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = SIN(Zeta#) * CosZeta#
        END IF
    END SELECT
END SELECT
tprob# = .5 * (1 + A#)
END FUNCTION

```

```
FUNCTION VVariance! (Var!())
```

```
  M! = SMEAN!(Var!())
```

```
  S! = 0
```

```
  FOR Number& = 1 TO UBOUND(Var!)
```

```
    S! = S! + (Var!(Number&) - M!) ^ 2
```

```
  NEXT Number&
```

```
  VVariance! = S! / UBOUND(Var!)
```

```
END FUNCTION
```

```
FUNCTION Zscore# (Z#)
```

```
' The formula used from Abramowitz & Stegun (1972) formula 2
```

```
IF Z# >= -14.14 AND Z# <= 14.14 THEN
```

```
  Y# = (.7071067812#) * Z#
```

```
ELSE
```

```
  Y# = 10
```

```
END IF
```

```
A1# = 7.052307839999999D-02
```

```
A2# = .0422820123#
```

```
A3# = .0092705272#
```

```
A4# = .0001520143#
```

```
A5# = .0002765672#
```

```
A6# = .0000430638#
```

```
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y#^2) + A3# * (Y#^3) + A4# * (Y#^4) + A5# * (Y#^5) + A6# * (Y#^6)) ^ (-16)
```

```
END FUNCTION
```

โปรแกรม LMMS-11.BAS ใช้คำนวณและแสดงผลค่าคะแนนเฉลี่ย และความแปรปรวนคะแนนจากการสุ่มแบบเมตริกซ์เชิง

พหุคูณ

'Program Name : LMMS-11.BAS

TYPE Fstr2

Round AS INTEGER

Group AS INTEGER

SumE AS SINGLE

SumE2 AS SINGLE

SumI AS SINGLE

SumI2 AS SINGLE

SumX2 AS SINGLE

END TYPE

DIM SHARED File2 AS Fstr2

CLS

N& = 1200: M& = 480

DO

CLS

PRINT "Result-MMS process to print..."

PRINT "You want to PRINT (1) Normal (2) Negative skewness (3) Positive skewness ? ";

ANS1\$ = INPUT\$(1)

ANS1\$ = UCASE\$(ANS1\$)

LOOP UNTIL INSTR(1, "123", ANS1\$) > 0

PRINT ANS1\$

DO

PRINT "You want to PRINT k = ? ";

k% = VAL(INPUT\$(1))

LOOP UNTIL k% = 2 OR k% = 4 OR k% = 6

PRINT k%

FileNo% = 2

OPEN "O", FileNo%, "LMMS-" + ANS1\$ + LTRIM\$(STR\$(k%)) + ".OUT"

WIDTH #FileNo%, 255

ns& = N& / k%: ms& = M& / k%

SumMean! = 0

SumVarE! = 0

SumVarI! = 0

SumVarEI! = 0

SumMean1! = 0

SumVarE1! = 0

SumVarI1! = 0

SumVarEI1! = 0

SumMean2! = 0

SumVarE2! = 0

SumVarI2! = 0

SumVarEI2! = 0

OPEN "R", #1, "MMS" + LTRIM\$(RTRIM\$(STR\$(ns&))) + "-" + ANS1\$ + ".DAT", LEN(File2)

GOSUB Head

Max% = 56 * k%

FOR I% = 1 TO LOF(1) / LEN(File2) STEP Max% * 2

FOR x% = 1 TO Max%

GET #1, I% + (x% - 1), File2

A! = (File2.SumE2 / ms& - ((File2.SumE)^2) / (ns& * ms&)) / (ns& - 1)

B! = (File2.SumI2 / ns& - ((File2.SUMI)^2) / (ns& * ns&)) / (ms& - 1)

C! = (File2.SumX2 - File2.SumE2 / ms& - File2.SumI2 / ns& + ((File2.SumE)^2 / (ms& * ns&))) / ((ns& - 1) * (ms& - 1))

Mean! = File2.SumE / (ns& * ms&)

VarE! = ((N& - 1) / N&) * ((A! - (1 - ms& / M&) * C) / ms&)

VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)

VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C

SumMean1! = SumMean1! + Mean!

SumVarE1! = SumVarE1! + VarE!

```

SumVarI1! = SumVarI1! + VarI!
SumVarEI1! = SumVarEI1! + VarEI!
IF File2.Group = 1 THEN
    RR1% = File2.Round
END IF
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
    'I% + Max% + (x% - 1) <= (1000 * k%)
    GET #1, I% + Max% + (x% - 1), File2
    A! = (File2.SumE2 / ns& - ((File2.SumE) ^ 2) / (ns& * ns&)) / (ns& - 1)
    B! = (File2.SumI2 / ns& - ((File2.SumI) ^ 2) / (ns& * ns&)) / (ns& - 1)
    C! = (File2.SumX2 - File2.SumE2 / ns& - File2.SumI2 / ns& + ((File2.SumE) ^ 2) / (ns& * ns&)) / ((ns& - 1) * (ns& - 1))
    Mean! = File2.SumE / (ns& * ns&)
    VarE! = ((N& - 1) / N&) * ((A! - (1 - ns& / M&) * C) / ns&)
    VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)
    VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C
    SumMean2! = SumMean2! + Mean!
    SumVarE2! = SumVarE2! + VarE!
    SumVarI2! = SumVarI2! + VarI!
    SumVarEI2! = SumVarEI2! + VarEI!
    IF File2.Group = 1 THEN
        RR2% = File2.Round
    END IF
    IF File2.Group = k% THEN
        Mean1! = SumMean1! / k%
        VarE1! = SumVarE1! / k%
        VarI1! = SumVarI1! / k%
        VarEI1! = SumVarEI1! / k%
        VarMean1! = ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI1!
        Mean2! = SumMean2! / k%
        VarE2! = SumVarE2! / k%
        VarI2! = SumVarI2! / k%
        VarEI2! = SumVarEI2! / k%
        VarMean2! = ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI2!
        SumMean! = SumMean1! + Mean1! + Mean2!
        SumVarE! = SumVarE1! + VarE1! + VarE2!
        SumVarI! = SumVarI1! + VarI1! + VarI2!
        SumVarEI! = SumVarEI1! + VarEI1! + VarEI2!
        SumVarMean! = SumVarMean1! + VarMean1! + VarMean2!
        PRINT #FileNo%, TAB(1); USING "#,###"; RR1%;
        PRINT #FileNo%, TAB(8); "Avg";
        PRINT #FileNo%, TAB(14); USING "##### "; Mean1!; VarE1!; VarI1!; VarEI1!;
        PRINT #FileNo%, USING "#####"; SQR(VarMean1!);
        PRINT #FileNo%, TAB(1 + 55); USING "#,###"; RR2%;
        PRINT #FileNo%, TAB(8 + 55); "Avg";
        PRINT #FileNo%, TAB(14 + 55); USING "##### "; Mean2!; VarE2!; VarI2!; VarEI2!;
        PRINT #FileNo%, USING "#####"; SQR(VarMean2!)
        SumMean1! = 0
        SumVarE1! = 0
        SumVarI1! = 0
        SumVarEI1! = 0
        SumMean2! = 0
        SumVarE2! = 0
        SumVarI2! = 0
        SumVarEI2! = 0
    END IF
ELSE
    IF File2.Group = k% THEN
        Mean1! = SumMean1! / k%

```

```

VarEI! = SumVarEI! / k%
VarI! = SumVarI! / k%
VarEI! = SumVarEI! / k%
VarMean! = ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI!
PRINT #FileNo%, TAB(1); USING "#,###"; RR1%;
PRINT #FileNo%, TAB(8); "Avg";
PRINT #FileNo%, TAB(14); USING "##### "; Mean!; VarEI!; VarI!; VarEI!;
PRINT #FileNo%, USING "#####"; SQR(VarMean!);
SumMean! = SumMean! + Mean!
SumVarE! = SumVarE! + VarEI!
SumVarI! = SumVarI! + VarI!
SumVarEI! = SumVarEI! + VarEI!
SumVarMean! = SumVarMean! + VarMean!
SumMean! = 0
SumVarE! = 0
SumVarI! = 0
SumVarEI! = 0
END IF
END IF
NEXT x%
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
    PRINT #FileNo%, CHR$(12)
    GOSUB Head
END IF
NEXT I%
PRINT #FileNo%,
PRINT #FileNo%, "-----"
Mean! = SumMean! / 1000
VarE! = SumVarE! / 1000
VarI! = SumVarI! / 1000
VarEI! = SumVarEI! / 1000
VarMean! = SumVarMean! / 1000
PRINT #FileNo%, TAB(8); "TOT";
PRINT #FileNo%, TAB(14); USING "##### "; Mean!; VarE!; VarI!; VarEI!;
PRINT #FileNo%, USING "#####"; SQR(VarMean!)
PRINT #FileNo%, "-----"
CLOSE
END
Head:
PRINT #FileNo%, TAB(1); "Result MMS process for ";
SELECT CASE ANS1$
CASE "1"
    PRINT #FileNo%, "Normal ";
CASE "2"
    PRINT #FileNo%, "Negative skewness ";
CASE ELSE
    PRINT #FileNo%, "Positive skewness ";
END SELECT
PRINT #FileNo%, USING "N = #,###"; N&;
PRINT #FileNo%, USING "M = #,###"; M&;
PRINT #FileNo%, USING "k = #"; k%;
PRINT #FileNo%, USING "n = ###"; ns&;
PRINT #FileNo%, USING "m = ###"; ms&
PRINT #FileNo%, "-----"
PRINT #FileNo%, "Round Matrix Mean VarE VarI VarEI S.d.Mean Round Matrix Mean VarE
VarI VarEI S.d.Mean"
'print "#,### # ##### ##### ##### #####"
PRINT #FileNo%, "-----"
RETURN

```

โปรแกรม LMMS-12.BAS ใช้แสดงผลเปรียบเทียบค่าคะแนนเฉลี่ย และความแปรปรวนคะแนนจากการสุ่มแบบเมตริกซ์

เชิงพหุคูณกับประชากร

'Program Name : LMMS-12.BAS

DECLARE FUNCTION Zscore# (Z#)

DECLARE FUNCTION tprob# (t#, k&)

***** LMMS-12.BAS *****

TYPE Fstr2

Round AS INTEGER

Group AS INTEGER

SumE AS SINGLE

SumE2 AS SINGLE

SUMI AS SINGLE

SumI2 AS SINGLE

SumX2 AS SINGLE

END TYPE

DIM SHARED File2 AS Fstr2

CLS

N& = 1200: M& = 480

DO

CLS

PRINT "Result-MMS process to PRINT..."

PRINT "You want to PRINT (1) Normal (2) Negative skewness (3) Positive skewness ? ";

ANS1\$ = INPUT\$(1)

ANS1\$ = UCASE\$(ANS1\$)

LOOP UNTIL INSTR(1, "123", ANS1\$) > 0

PRINT ANS1\$

DO

PRINT "You want to PRINT k = ? ";

k% = VAL(INPUT\$(1))

LOOP UNTIL k% = 2 OR k% = 4 OR k% = 6

PRINT k%

OPEN "I", #1, "MEAN.DAT"

SELECT CASE ANS1\$

CASE "1"

LINE INPUT #1, NNS

Mn1! = VAL(LEFT\$(NNS, 9))

Var1! = VAL(RIGHT\$(NNS, 15))

CASE "2"

LINE INPUT #1, NNS

LINE INPUT #1, NNS

Mn1! = VAL(LEFT\$(NNS, 9))

Var1! = VAL(RIGHT\$(NNS, 15))

CASE ELSE

LINE INPUT #1, NNS

LINE INPUT #1, NNS

LINE INPUT #1, NNS

Mn1! = VAL(LEFT\$(NNS, 9))

Var1! = VAL(RIGHT\$(NNS, 15))

END SELECT

CLOSE #1

FileNo% = 2

OPEN "O", FileNo%, "LMMS" + ANS1\$ + LTRIM\$(STR\$(k%)) + ".OUT"

WIDTH #FileNo%, 255

ns& = N& / k%: ms& = M& / k%

SumMean! = 0

SumVarE! = 0

SumVarI! = 0

```

SumVarEI! = 0
SumMean1! = 0
SumVarE1! = 0
SumVarI1! = 0
SumVarEI1! = 0
SumMean2! = 0
SumVarE2! = 0
SumVarI2! = 0
SumVarEI2! = 0
OPEN "R", #1, "MMS" + LTRIM$(RTRIM$(STR$(ns&))) + "-" + ANSI$ + ".DAT", LEN(File2)
GOSUB Head
Max% = 56 * k%
FOR I% = 1 TO LOF(1) / LEN(File2) STEP Max% * 2
  FOR x% = 1 TO Max%
    GET #1, I% + (x% - 1), File2
    Round1% = File2.Round
    A! = (File2.SumE2 / ms& - ((File2.SumE) ^ 2) / (ns& * ms&)) / (ns& - 1)
    B! = (File2.SumI2 / ns& - ((File2.SUMI) ^ 2) / (ns& * ms&)) / (ms& - 1)
    C! = (File2.SumX2 - File2.SumE2 / ms& - File2.SumI2 / ns& + ((File2.SumE) ^ 2 / (ms& * ns&))) /
    ((ns& - 1) * (ms& - 1))
    Mean! = File2.SumE / (ns& * ms&)
    VarE! = ((N& - 1) / N&) * ((A! - (1 - ms& / M&) * C) / ms&)
    VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)
    VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C
    SumMean1! = SumMean1! + Mean!
    SumVarE1! = SumVarE1! + VarE!
    SumVarI1! = SumVarI1! + VarI!
    SumVarEI1! = SumVarEI1! + VarEI!
    IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
      GET #1, I% + Max% + (x% - 1), File2
      Round2% = File2.Round
      A! = (File2.SumE2 / ms& - ((File2.SumE) ^ 2) / (ns& * ms&)) / (ns& - 1)
      B! = (File2.SumI2 / ns& - ((File2.SUMI) ^ 2) / (ns& * ms&)) / (ms& - 1)
      C! = (File2.SumX2 - File2.SumE2 / ms& - File2.SumI2 / ns& + ((File2.SumE) ^ 2 / (ms& * ns&))) /
      ((ns& - 1) * (ms& - 1))
      Mean! = File2.SumE / (ns& * ms&)
      VarE! = ((N& - 1) / N&) * ((A! - (1 - ms& / M&) * C) / ms&)
      VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)
      VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C
      SumMean2! = SumMean2! + Mean!
      SumVarE2! = SumVarE2! + VarE!
      SumVarI2! = SumVarI2! + VarI!
      SumVarEI2! = SumVarEI2! + VarEI!
      IF File2.Group = k% THEN
        Mean1! = M& * SumMean1! / k%
        VarE1! = (M& ^ 2) * SumVarE1! / k%
        VarI1! = (N& ^ 2) * SumVarI1! / k%
        VarEI1! = (M& ^ 2) * SumVarEI1! / k%
        VarMean1! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI1!
        Mean2! = M& * SumMean2! / k%
        VarE2! = (M& ^ 2) * SumVarE2! / k%
        VarI2! = (N& ^ 2) * SumVarI2! / k%
        VarEI2! = (M& ^ 2) * SumVarEI2! / k%
        VarMean2! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI2!
        SumMean! = SumMean1! + Mean1! + Mean2!
        SumVarE! = SumVarE1! + VarE1! + VarE2!
        SumVarI! = SumVarI1! + VarI1! + VarI2!

```

```

SumVarEI! = SumVarEI! + VarEI1! + VarEI2!
SumVarMean! = SumVarMean! + VarMean1! + VarMean2!
PRINT #FileNo%, TAB(1); USING "#,###"; Round1%;
PRINT #FileNo%, TAB(8); USING "###.####"; Mean1!;
PRINT #FileNo%, TAB(17); USING "#,###,###.####"; VarEI!;
PRINT #FileNo%, TAB(32); USING "#,###.####"; SQR(VarMean1!);
t# = ((Mean1! - Mn1!) / (SQR(VarEI!) / SQR(ns&)))
tp# = ABS(tprob#(t#, ns& - 1))
PRINT #FileNo%, TAB(44); USING "###.####"; t#;
IF tp# > .995 THEN
    PRINT #FileNo%, "***";
ELSE
    IF tp# > .975 THEN
        PRINT #FileNo%, "**";
    END IF
END IF
PRINT #FileNo%, TAB(54); USING "#,###.####"; ((ns& - 1) * (VarEI! / Var1!));

PRINT #FileNo%, TAB(1 + 66); USING "#,###"; Round2%;
PRINT #FileNo%, TAB(8 + 66); USING "###.####"; Mean2!;
PRINT #FileNo%, TAB(17 + 66); USING "#,###,###.####"; VarE2!;
PRINT #FileNo%, TAB(32 + 66); USING "#,###.####"; SQR(VarMean2!);
t# = ((Mean2! - Mn1!) / (SQR(VarE2!) / SQR(ns&)))
PRINT #FileNo%, TAB(44 + 66); USING "###.####"; t#;
tp# = ABS(tprob#(t#, ns& - 1))
IF tp# > .995 THEN
    PRINT #FileNo%, "***";
ELSE
    IF tp# > .975 THEN
        PRINT #FileNo%, "**";
    END IF
END IF
PRINT #FileNo%, TAB(54 + 66); USING "#,###.####"; ((ns& - 1) * (VarE2! / Var1!));
SumMean1! = 0
SumVarEI! = 0
SumVar1! = 0
SumVarEI1! = 0
SumMean2! = 0
SumVarE2! = 0
SumVar2! = 0
SumVarEI2! = 0
END IF
ELSE
    IF File2.Group = k% THEN
        Mean1! = M& * SumMean1! / k%
        VarEI! = (M& ^ 2) * SumVarEI! / k%
        Var1! = (N& ^ 2) * SumVar1! / k%
        VarEI1! = (M& ^ 2) * SumVarEI1! / k%
        VarMean1! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI1!
        PRINT #FileNo%, TAB(1); USING "#,###"; Round1%;
        PRINT #FileNo%, TAB(8); USING "###.####"; Mean1!;
        PRINT #FileNo%, TAB(17); USING "#,###,###.####"; VarEI!;
        PRINT #FileNo%, TAB(32); USING "#,###.####"; SQR(VarMean1!);
        t# = ((Mean1! - Mn1!) / (SQR(VarEI!) / SQR(ns&)))
        PRINT #FileNo%, TAB(44); USING "###.####"; t#;
        tp# = ABS(tprob#(t#, ns& - 1))
        IF tp# > .995 THEN

```

```

        PRINT #FileNo%, "****";
    ELSE
        IF tp# > .975 THEN
            PRINT #FileNo%, "****";
        END IF
    END IF
    PRINT #FileNo%, TAB(54); USING "#,###.####", ((ns& - 1) * (VarE! / VarI!))
    SumMean! = SumMean! + MeanI!
    SumVarE! = SumVarE! + VarE!
    SumVarI! = SumVarI! + VarI!
    SumVarEI! = SumVarEI! + VarEI!
    SumVarMean! = SumVarMean! + VarMeanI!
    SumMeanI! = 0
    SumVarEI! = 0
    SumVarI! = 0
    SumVarEI! = 0
END IF
END IF
NEXT x%
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
    PRINT #FileNo%, CHR$(12)
    GOSUB Head
END IF
NEXT I%
PRINT #FileNo%, "-----"
Mean! = SumMean! / 1000
VarE! = SumVarE! / 1000
VarI! = SumVarI! / 1000
VarEI! = SumVarEI! / 1000
VarMean! = SumVarMean! / 1000
PRINT #FileNo%, TAB(1); "TOTAL";
PRINT #FileNo%, TAB(8); USING "###.####", Mean!;
PRINT #FileNo%, TAB(17); USING "#,###.####", VarE!;
PRINT #FileNo%, TAB(32); USING "#,###.####", SQR(VarMean!);
t# = ((Mean! - MnI!) / (SQR(VarE!) / SQR(ns&)))
PRINT #FileNo%, TAB(44); USING "#,###.####", t#;
tp# = ABS(tprob#(t#, ns& - 1))
IF tp# > .995 THEN
    PRINT #FileNo%, "****";
ELSE
    IF tp# > .975 THEN
        PRINT #FileNo%, "****";
    END IF
END IF
PRINT #FileNo%, TAB(54); USING "#,###.####", ((ns& - 1) * (VarE! / VarI!))
PRINT #FileNo%, "-----"
PRINT #FileNo%, CHR$(12)
CLOSE
END
Head:
    PRINT #FileNo%, TAB(1); "Result MMS process for ";
    SELECT CASE ANSI$
    CASE "1"
        PRINT #FileNo%, "Normal ";
    CASE "2"

```

```

PRINT #FileNo%, "Negative skewness ";
CASE ELSE
PRINT #FileNo%, "Positive skewness ";
END SELECT
PRINT #FileNo%, USING "N = ####", N&;
PRINT #FileNo%, USING "M = ####", M&;
PRINT #FileNo%, USING "k = #", k%;
PRINT #FileNo%, USING "n = ####", ns&;
PRINT #FileNo%, USING "m = ####", ms&;
PRINT #FileNo%, USING "With mean = ####.####", Mn1!;
PRINT #FileNo%, USING "Variance = ####.####", Var1!
PRINT #FileNo%, "-----"

PRINT #FileNo%, "Round  Mean  Variance  S.E.Mean  t-value  Chi-square  Round  Mean
Variance  S.E.Mean  t-value  Chi-square"
PRINT #FileNo%, "#.###  ###.####  ###.###.####  ###.####  ###.####  ###.####  ###.###
###.###  ###.###.####  ###.###.####  ###.####  ###.####  ###.###"
PRINT #FileNo%, "-----"

RETURN

FUNCTION tprob# (t#, k&)
'parameter tprob = probability when t<=t#
' t# = t-score value to find prob.
' k& = degree of freedom
'
' if k&>30 then limit of t-score to Z-score
IF k& > 30 THEN
tprob# = Zscore#(t#)
EXIT FUNCTION
END IF
' This function calculated by seeing Abramowitz & Stegun (1972)
Zeta# = ATN(t# / SQR(k&))
SELECT CASE k&
CASE 1
A# = 2 * Zeta# / pi
CASE ELSE
odd& = ((k& MOD 2) = 1)
SELECT CASE odd&
CASE -1 'odd
IF k& = 3 THEN
A# = 2 / pi * (Zeta# + SIN(Zeta#) * COS(Zeta#))
ELSE
CosZeta# = COS(Zeta#)
FOR x& = 2 TO k& - 3 STEP 2
TX# = 1
FOR Y& = 2 TO x& STEP 2
TX# = TX# * Y& / (Y& + 1)
NEXT Y&
TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
CosZeta# = CosZeta# + TX#
NEXT x&
A# = 2 / pi * (Zeta# + SIN(Zeta#) * CosZeta#)
END IF
CASE ELSE 'even
IF k& = 2 THEN
A# = SIN(Zeta#)

```

```

ELSE
  CosZeta# = 1
  FOR x& = 1 TO k& - 3 STEP 2
    TX# = 1
    FOR Y& = 1 TO x& STEP 2
      TX# = TX# * Y& / (Y& + 1)
    NEXT Y&
    TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
    CosZeta# = CosZeta# + TX#
  NEXT x&
  A# = SIN(Zeta#) * CosZeta#
END IF
END SELECT
END SELECT
tprob# = .5 * (1 + A#)
END FUNCTION

FUNCTION Zscore# (Z#)
' The formula used from Abramowitz & Stegun (1972) formula 2

IF Z# >= -14.14 AND Z# <= 14.14 THEN
  Y# = (.7071067812#) * Z#
ELSE
  Y# = 10
END IF
A1# = 7.052307839999999D-02
A2# = .0422820123#
A3# = .0092705272#
A4# = .0001520143#
A5# = .0002765672#
A6# = .0000430638#
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) +
A6# * (Y# ^ 6)) ^ (-16)
END FUNCTION

```

โปรแกรม LMMS-SRS.BAS ใช้แสดงผลเปรียบเทียบค่าคะแนนเฉลี่ย และความแปรปรวนคะแนนระหว่าง

'การสุ่มแบบเมตริกซ์และการสุ่มแบบง่าย

***** LMMS-SRS.BAS *****

DECLARE FUNCTION Zscore# (Z#)

DECLARE FUNCTION tprob# (t#, k&)

'\$DYNAMIC

TYPE Fstr2

Round AS INTEGER

Group AS INTEGER

SumE AS SINGLE

SumE2 AS SINGLE

SumI AS SINGLE

SumI2 AS SINGLE

SumX2 AS SINGLE

END TYPE

TYPE Fstr

MMSMean AS SINGLE

MMSVar AS SINGLE

SRSMean AS SINGLE

SRSVar AS SINGLE

END TYPE

TYPE Fstr3

Round AS INTEGER

Group AS INTEGER

SumX AS SINGLE

SumX2 AS SINGLE

END TYPE

DIM SHARED File2 AS Fstr2

DIM SHARED File3 AS Fstr3

DIM SHARED VT(1 TO 1001) AS Fstr

CLS

N& = 1200: M& = 480

DO

CLS

PRINT "Result-MMS process to PRINT..."

PRINT "You want to PRINT (1) Normal (2) Negative skewness (3) Positive skewness ? ";

Ans1\$ = INPUT\$(1)

Ans1\$ = UCASE\$(Ans1\$)

LOOP UNTIL INSTR(1, "123", Ans1\$) > 0

PRINT Ans1\$

DO

PRINT "You want to PRINT k = ? ";

k% = VAL(INPUT\$(1))

LOOP UNTIL k% = 2 OR k% = 4 OR k% = 6

PRINT k%

ns& = N& / k%: ms& = M& / k%

***** MMS Process *****

SumMean! = 0

SumVarE! = 0

SumVarI! = 0

SumVarEI! = 0

SumMean1! = 0

SumVarE1! = 0

SumVarI1! = 0

SumVarEI1! = 0

SumMean2! = 0

```

SumVarE2! = 0
SumVarI2! = 0
SumVarEI2! = 0
Sig05% = 0: Sig01% = 0
OPEN "R", #1, "MMS" + LTRIM$(RTRIM$(STR$(ns&))) + "-" + Ans1$ + ".DAT", LEN(File2)
PRINT "MMS" + LTRIM$(RTRIM$(STR$(ns&))) + "-" + Ans1$ + ".DAT V.S.";
Max% = 56 * k%
FOR I% = 1 TO LOF(1) / LEN(File2) STEP Max% * 2
  FOR x% = 1 TO Max%
    GET #1, I% + (x% - 1), File2
    Round1% = File2.Round
    A! = (File2.SumE2 / ms& - ((File2.SumE) ^ 2) / (ns& * ms&)) / (ns& - 1)
    B! = (File2.SumI2 / ns& - ((File2.SUMI) ^ 2) / (ns& * ms&)) / (ms& - 1)
    C! = (File2.SumX2 - File2.SumE2 / ms& - File2.SumI2 / ns& + ((File2.SumE) ^ 2 / (ms& * ns&))) /
    ((ns& - 1) * (ms& - 1))
    Mean! = File2.SumE / (ns& * ms&)
    VarE! = ((N& - 1) / N&) * ((A! - (1 - ms& / M&) * C) / ms&)
    VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)
    VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C
    SumMean1! = SumMean1! + Mean!
    SumVarE1! = SumVarE1! + VarE!
    SumVarI1! = SumVarI1! + VarI!
    SumVarEI1! = SumVarEI1! + VarEI!
    IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
      GET #1, I% + Max% + (x% - 1), File2
      Round2% = File2.Round
      A! = (File2.SumE2 / ms& - ((File2.SumE) ^ 2) / (ns& * ms&)) / (ns& - 1)
      B! = (File2.SumI2 / ns& - ((File2.SUMI) ^ 2) / (ns& * ms&)) / (ms& - 1)
      C! = (File2.SumX2 - File2.SumE2 / ms& - File2.SumI2 / ns& + ((File2.SumE) ^ 2 / (ms& * ns&))) / ((ns& - 1) * (ms& - 1))
      Mean! = File2.SumE / (ns& * ms&)
      VarE! = ((N& - 1) / N&) * ((A! - (1 - ms& / M&) * C) / ms&)
      VarI! = ((M& - 1) / M&) * ((B! - (1 - ns& / N&) * C) / ns&)
      VarEI! = (((N& - 1) * (M& - 1)) / (N& * M&)) * C
      SumMean2! = SumMean2! + Mean!
      SumVarE2! = SumVarE2! + VarE!
      SumVarI2! = SumVarI2! + VarI!
      SumVarEI2! = SumVarEI2! + VarEI!
      IF File2.Group = k% THEN
        Mean1! = M& * SumMean1! / k%
        VarE1! = (M& ^ 2) * SumVarE1! / k%
        VarI1! = (N& ^ 2) * SumVarI1! / k%
        VarEI1! = (M& ^ 2) * SumVarEI1! / k%
        VarMean1! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI1!
        Mean2! = M& * SumMean2! / k%
        VarE2! = (M& ^ 2) * SumVarE2! / k%
        VarI2! = (N& ^ 2) * SumVarI2! / k%
        VarEI2! = (M& ^ 2) * SumVarEI2! / k%
        VarMean2! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI2!
        SumMean! = SumMean! + Mean1! + Mean2!
        SumVarE! = SumVarE! + VarE1! + VarE2!
        SumVarI! = SumVarI! + VarI1! + VarI2!
        SumVarEI! = SumVarEI! + VarEI1! + VarEI2!
        SumVarMean! = SumVarMean! + VarMean1! + VarMean2!
        VT(Round1%).MMSMean = Mean1!
        VT(Round1%).MMSVar = VarE1!
        VT(Round2%).MMSMean = Mean2!
        VT(Round2%).MMSVar = VarE2!

```

```

SumMean1! = 0
SumVarE1! = 0
SumVarI1! = 0
SumVarEI1! = 0
SumMean2! = 0
SumVarE2! = 0
SumVarI2! = 0
SumVarEI2! = 0
END IF
ELSE
  IF File2.Group = k% THEN
    Mean1! = M& * SumMean1! / k%
    VarE1! = (M& ^ 2) * SumVarE1! / k%
    VarI1! = (N& ^ 2) * SumVarI1! / k%
    VarEI1! = (M& ^ 2) * SumVarEI1! / k%
    VarMean1! = (M& ^ 2) * ((k% - 1) / ((N& - 1) * (M& - 1))) * VarEI1!
    VT(Round1%).MMSMean = Mean1!
    VT(Round1%).MMSVar = VarE1!
    SumMean! = SumMean! + Mean1!
    SumVarE! = SumVarE! + VarE1!
    SumVarI! = SumVarI! + VarI1!
    SumVarEI! = SumVarEI! + VarEI1!
    SumVarMean! = SumVarMean! + VarMean1!
    SumMean1! = 0
    SumVarE1! = 0
    SumVarI1! = 0
    SumVarEI1! = 0
  END IF
END IF
NEXT x%
NEXT I%
Mean! = SumMean! / 1000
VarE! = SumVarE! / 1000
VarI! = SumVarI! / 1000
VarEI! = SumVarEI! / 1000
VarMean! = SumVarMean! / 1000
VT(1001).MMSMean = Mean!
VT(1001).MMSVar = VarE!
CLOSE

***** SRS PROCESS *****

OPEN "R", #1, "S" + LTRIM$(RTRIM$(STR$(ns&))) + "1-" + Ans1$ + ".DAT", LEN(File3)
PRINT "S" + LTRIM$(RTRIM$(STR$(ns&))) + "1-" + Ans1$ + ".DAT"
SumMean! = 0
SumVar! = 0
Max% = 56 * k%
Max% = Max% - (Max% MOD k%)
FOR I% = 1 TO LOF(1) / LEN(File3) STEP Max% * 2
  FOR x% = 1 TO Max%
    GET #1, I% + (x% - 1), File3
    IF File3.Group = 1 THEN
      Mean! = File3.SumX / ns&
      Variance! = (File3.SumX2 - (File3.SumX ^ 2 / ns&)) / (ns& - 1)
      SumMean! = SumMean! + Mean!
      SumVar! = SumVar! + Variance!
      VT(File3.Round).SRSMean = Mean!
    END IF
  NEXT x%
NEXT I%

```

```

VT(File3.Round).SRSVar = Variance!
IF (I% + Max% + (x% - 1)) <= (1000 * k%) THEN
  GET #1, I% + Max% + (x% - 1), File3
  IF File3.Group = 1 THEN
    Mean! = File3.SumX / ns&
    Variance! = (File3.SumX2 - (File3.SumX ^ 2 / ns&)) / (ns& - 1)
    SumMean! = SumMean! + Mean!
    SumVar! = SumVar! + Variance!
    VT(File3.Round).SRSMean = Mean!
    VT(File3.Round).SRSVar = Variance!
  END IF
END IF
END IF
NEXT x%
NEXT I%
Mean! = SumMean! / 1000
Variance! = SumVar! / 1000
VT(1001).SRSMean = Mean!
VT(1001).SRSVar = Variance!
CLOSE #1
***** Print
GOSUB Head
FOR I% = 1 TO 1000 STEP 112
  FOR j% = I% TO (I% + 55)
    IF j% = 1 THEN
      MaxMean# = ABS(VT(j%).SRSMean - VT(j%).MMSMean)
      MaxVar# = ABS(VT(j%).SRSVar - VT(j%).MMSVar)
      MinMean# = ABS(VT(j%).SRSMean - VT(j%).MMSMean)
      MinVar# = ABS(VT(j%).SRSVar - VT(j%).MMSVar)
    ELSE
      IF ABS(VT(j%).SRSMean - VT(j%).MMSMean) > MaxMean# THEN
        MaxMean# = ABS(VT(j%).SRSMean - VT(j%).MMSMean)
      END IF
      IF ABS(VT(j%).SRSMean - VT(j%).MMSMean) < MinMean# THEN
        MinMean# = ABS(VT(j%).SRSMean - VT(j%).MMSMean)
      END IF
      IF ABS(VT(j%).SRSVar - VT(j%).MMSVar) > MaxVar# THEN
        MaxVar# = ABS(VT(j%).SRSVar - VT(j%).MMSVar)
      END IF
      IF ABS(VT(j%).SRSVar - VT(j%).MMSVar) < MinVar# THEN
        MinVar# = ABS(VT(j%).SRSVar - VT(j%).MMSVar)
      END IF
    END IF
    t# = (VT(j%).MMSMean - VT(j%).SRSMean) / SQR(VT(j%).MMSVar / ns& + VT(j%).SRSVar / ns&)
    tp# = ABS(tprob#(t#, ns& - 1))
    IF VT(j%).MMSVar > VT(j%).SRSVar THEN
      F# = VT(j%).MMSVar / VT(j%).SRSVar
    ELSE
      F# = VT(j%).SRSVar / VT(j%).MMSVar
    END IF
  END IF
  SELECT CASE k%
  CASE 2
    IF F# >= 1.145# THEN
      Sig05% = Sig05% + 1
    END IF
    IF F# >= 1.21# THEN
      Sig01% = Sig01% + 1
    END IF
  
```

```

END IF
CASE 4
  IF F# >= 1.21# THEN
    Sig05% = Sig05% + 1
  END IF
  IF F# >= 1.31# THEN
    Sig01% = Sig01% + 1
  END IF
CASE ELSE
END SELECT
PRINT TAB(1); USING "#,### ##.## ##.##,###.## ##.##,###.## #,###.##"; j%, VT(j%).
MMSMean; VT(j%).MMSVar; VT(j%).SRSMean; VT(j%).SRSVar; t#;
IF tp# > .995 THEN
  PRINT "***";
ELSE
  IF tp# > .975 THEN
    PRINT "**";
  END IF
END IF
PRINT TAB(59); USING "#,###.##"; F#;
kk% = j% + 56
IF kk% <= 1000 THEN
  t# = (VT(kk%).MMSMean - VT(kk%).SRSMean) / SQRT(VT(kk%).MMSVar / ns& + VT(kk%).SRSVar / ns&)
  tp# = ABS(tprob#(t#, ns& - 1))
  IF VT(kk%).MMSVar > VT(kk%).SRSVar THEN
    F# = VT(kk%).MMSVar / VT(kk%).SRSVar
  ELSE
    F# = VT(kk%).SRSVar / VT(kk%).MMSVar
  END IF
END IF
SELECT CASE k%
CASE 2
  IF F# >= 1.145# THEN
    Sig05% = Sig05% + 1
  END IF
  IF F# >= 1.21# THEN
    Sig01% = Sig01% + 1
  END IF
CASE 4
  IF F# >= 1.21# THEN
    Sig05% = Sig05% + 1
  END IF
  IF F# >= 1.31# THEN
    Sig01% = Sig01% + 1
  END IF
CASE ELSE
END SELECT
PRINT TAB(71); USING "#,### ##.## ##.##,###.## ##.##,###.## #,###.##"; kk%; VT
(kk%).MMSMean; VT(kk%).MMSVar; VT(kk%).SRSMean; VT(kk%).SRSVar; t#;
IF tp# > .995 THEN
  PRINT "***";
ELSE
  IF tp# > .975 THEN
    PRINT "**";
  END IF
END IF
PRINT TAB(59 + 70); USING "#,###.##"; F#;
END IF

```

```

NEXT j%
IF kk% <= 1000 THEN
    PRINT CHR$(12)
    GOSUB Head
END IF
NEXT I%
PRINT
PRINT "-----"
j% = 1001
t# = (VT(j%).MMSMean - VT(j%).SRSMean) / SQR(VT(j%).MMSVar / ns& + VT(j%).SRSVar / ns&)
tp# = ABS(tprob#(t#, ns& - 1))
IF VT(j%).MMSVar > VT(j%).SRSVar THEN
    F# = VT(j%).MMSVar / VT(j%).SRSVar
ELSE
    F# = VT(j%).SRSVar / VT(j%).MMSVar
END IF
PRINT TAB(1); USING "TOTAL ###.## ###.###.## ###.## #,###.###.## #,###.##"; VT(j%).
MMSMean; VT(j%).MMSVar; VT(j%).SRSMean; VT(j%).SRSVar; t#;
IF tp# > .995 THEN
    PRINT "****";
ELSE
    IF tp# > .975 THEN
        PRINT "***";
    END IF
END IF
PRINT TAB(59); USING "#,###.##"; F#
PRINT "-----"
PRINT CHR$(12)
PRINT k%
PRINT USING "####"; Sig05%; Sig01%
PRINT USING "Diff. Mean Min-Max : ###.#### ###.####"; MinMean#; MaxMean#
PRINT USING "Diff. Var Min-Max : ##,###.#### ##,###.####"; MinVar#; MaxVar#
END
Head:
WIDTH "LPT1:"; 255
PRINT TAB(1); "Compare SRS & MMS process for ";
SELECT CASE Ans1$
CASE "1"
    PRINT "Normal ";
CASE "2"
    PRINT "Negative skewness ";
CASE ELSE
    PRINT "Positive skewness ";
END SELECT
PRINT USING "N = #,####"; N&;
PRINT USING "M = #,####"; M&;
PRINT USING "k = #"; k%;
PRINT USING "n = ####"; ns&;
PRINT USING "m = ####"; ms&
PRINT "-----"
PRINT "      MMS      SRS      MMS      SRS      "
PRINT "Round Mean  Variance  Mean  Variance  t-value  F-value  Round Mean  variance  "
PRINT "Mean  variance  t-value  F-value  "

```

[illegible]

RETURN

```

REM $STATISTIC
FUNCTION tprob# (t#, k&)
'parameter tprob = probability when t<=t#
t# = t-score value to find prob.
k& = degree of freedom

'if k&>30 then limit of t-score to Z-score
IF k& > 30 THEN
    tprob# = Zscore#(t#)
EXIT FUNCTION
END IF

' This function calculated by seeing Abramowitz & Stegun (1972)
Zeta# = ATN(t# / SQR(k&))
SELECT CASE k&
CASE 1
    A# = 2 * Zeta# / pi
CASE ELSE
    odd& = ((k& MOD 2) = 1)
    SELECT CASE odd&
    CASE -1 'odd
        IF k& = 3 THEN
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * COS(Zeta#))
        ELSE
            CosZeta# = COS(Zeta#)
            FOR x& = 2 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 2 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = 2 / pi * (Zeta# + SIN(Zeta#) * CosZeta#)
        END IF
    CASE ELSE 'even
        IF k& = 2 THEN
            A# = SIN(Zeta#)
        ELSE
            CosZeta# = 1
            FOR x& = 1 TO k& - 3 STEP 2
                TX# = 1
                FOR Y& = 1 TO x& STEP 2
                    TX# = TX# * Y& / (Y& + 1)
                NEXT Y&
                TX# = TX# * ((COS(Zeta#)) ^ (x& + 1))
                CosZeta# = CosZeta# + TX#
            NEXT x&
            A# = SIN(Zeta#) * CosZeta#
        END IF
    END SELECT
END SELECT
END FUNCTION

```

```
tprob# = .5 * (1 + A#)
END FUNCTION
```

```
FUNCTION Zscore# (Z#)
```

```
' The formula used from Abramowitz & Stegun (1972 ) fomula 2
```

```
IF Z# >= -14.14 AND Z# <= 14.14 THEN
```

```
Y# = (.7071067812#) * Z#
```

```
ELSE
```

```
Y# = 10
```

```
END IF
```

```
A1# = 7.052307839999999D-02
```

```
A2# = .0422820123#
```

```
A3# = .0092705272#
```

```
A4# = .0001520143#
```

```
A5# = .0002765672#
```

```
A6# = .0000430638#
```

```
Zscore# = 1 - .5 * (1 + A1# * Y# + A2# * (Y# ^ 2) + A3# * (Y# ^ 3) + A4# * (Y# ^ 4) + A5# * (Y# ^ 5) +  
A6# * (Y# ^ 6)) ^ (-16)
```

```
END FUNCTION
```

ภาคผนวก ข

ตารางแสดงผลการดำเนินงาน

ภาคผนวก ข

ตาราง 23 แสดงค่าความสามารถของผู้สอบ (O_i) กรณีความสามารถของผู้สอบมีการแจกแจงแบบปกติ

คนที่	O _i	คนที่	O _i	คนที่	O _i	คนที่	O _i	คนที่	O _i	คนที่	O _i
1	-0.8472	56	-0.1005	111	-0.6130	166	-0.0411	221	-0.4679	276	-0.0437
2	-1.6865	57	-0.3223	112	-1.7997	167	0.1487	222	1.4174	277	-1.4823
3	0.7169	58	0.7129	113	-1.1171	168	-0.0620	223	1.5366	278	0.2250
4	0.8074	59	-1.5414	114	-0.9483	169	0.0622	224	-1.3645	279	0.9451
5	-1.2222	60	-1.5088	115	0.1609	170	0.0077	225	-0.4756	280	-0.3647
6	0.4737	61	-0.1967	116	-1.3784	171	-0.2527	226	0.8752	281	0.6776
7	-0.1749	62	1.9872	117	1.7441	172	-1.0993	227	-0.8308	282	1.0276
8	0.3660	63	-2.1751	118	0.3150	173	0.8223	228	-0.3118	283	0.7127
9	0.6106	64	-0.8433	119	-1.4392	174	0.1342	229	0.3970	284	-1.0065
10	-1.4499	65	-2.0092	120	-0.3146	175	1.1261	230	0.4344	285	-0.3871
11	1.7190	66	1.0986	121	1.4267	176	-0.8848	231	-2.8119	286	1.1847
12	-1.0001	67	0.2850	122	0.8185	177	0.5230	232	-0.3612	287	0.3527
13	-2.0516	68	0.1150	123	-0.6556	178	-1.3612	233	-1.5821	288	-0.0409
14	2.1323	69	0.6545	124	0.0504	179	-0.1442	234	0.7172	289	0.6659
15	0.0528	70	-1.3978	125	0.5333	180	1.8719	235	2.0417	290	0.6105
16	-1.1165	71	0.7015	126	-0.9012	181	1.9694	236	-1.4164	291	0.9582
17	-0.1697	72	0.3822	127	0.2273	182	0.1406	237	0.3957	292	-0.8593
18	0.9277	73	0.3963	128	-0.8881	183	0.2679	238	0.0491	293	2.1356
19	0.7814	74	-1.1315	129	-1.1117	184	2.2264	239	-1.1825	294	1.7624
20	-0.3701	75	-0.4396	130	1.2609	185	1.3622	240	0.4473	295	-0.5403
21	1.0546	76	1.2383	131	0.1864	186	0.1475	241	0.9667	296	-0.9958
22	-0.5810	77	1.1347	132	-0.7374	187	2.2056	242	1.2056	297	0.7362
23	-1.1545	78	1.6564	133	-0.6063	188	-2.2685	243	1.0957	298	-0.2329
24	-0.2174	79	-0.6111	134	0.3634	189	1.0594	244	-0.0328	299	1.5978
25	0.1798	80	0.4361	135	0.0933	190	0.6626	245	0.8735	300	0.5292
26	-0.6169	81	-0.6525	136	-0.4851	191	0.7619	246	1.1611	301	-1.0708
27	-0.4542	82	0.0519	137	-0.0876	192	1.5552	247	-0.5027	302	-1.2242
28	-1.0166	83	-0.3402	138	0.5475	193	0.0234	248	1.3299	303	-0.7649
29	-0.3449	84	-0.0720	139	1.6185	194	-0.3778	249	-0.9494	304	-0.4704
30	-0.5407	85	0.8054	140	0.2612	195	-1.6429	250	1.0062	305	0.8138
31	0.3727	86	-0.6543	141	1.5131	196	-0.7673	251	0.3121	306	-0.9328
32	-0.9480	87	0.5000	142	-0.6666	197	-2.1238	252	0.0560	307	0.1495
33	0.5876	88	0.5014	143	0.1883	198	0.3467	253	-0.4615	308	0.7219
34	-0.3539	89	0.3224	144	0.0224	199	-0.4325	254	2.6506	309	-0.8464
35	0.7237	90	-0.6880	145	-2.3467	200	-0.8288	255	0.2259	310	-2.0173
36	0.8544	91	0.1868	146	0.6243	201	-0.9188	256	-0.0352	311	1.2916
37	0.0366	92	-0.1730	147	0.4431	202	0.1260	257	0.8462	312	0.1128
38	0.9812	93	0.0919	148	0.0030	203	-0.2137	258	-1.0926	313	-0.9359
39	-1.4128	94	0.7434	149	-1.3842	204	0.3909	259	-0.6473	314	0.7494
40	0.2421	95	0.8071	150	-1.7013	205	0.9359	260	-0.9080	315	0.1379
41	-1.2672	96	0.2534	151	-0.1231	206	-0.2367	261	-0.0018	316	0.4665
42	-0.1115	97	1.3006	152	-0.8011	207	0.7842	262	1.1550	317	-0.6557
43	0.2819	98	-0.1890	153	0.6829	208	-0.9489	263	1.6402	318	-0.9083
44	1.0759	99	1.1207	154	-0.5774	209	-0.9043	264	0.0935	319	-1.7207
45	0.7637	100	-0.4901	155	-1.6058	210	0.3413	265	-1.0802	320	0.7523
46	-0.5249	101	-1.3051	156	0.2587	211	-0.5229	266	-1.7707	321	0.4523
47	-0.8279	102	0.4306	157	-0.7441	212	-0.3013	267	-0.9794	322	-0.3281
48	-0.5885	103	-0.8726	158	0.8903	213	-1.4482	268	0.2072	323	-0.5781
49	-0.8276	104	-1.6207	159	-0.9079	214	0.8021	269	-1.1697	324	2.5813
50	2.3748	105	-0.6085	160	1.8283	215	-0.9157	270	0.1161	325	1.6106
51	-0.0337	106	0.1416	161	-0.2244	216	-0.5118	271	0.5650	326	-0.0803
52	0.3827	107	0.1389	162	0.2529	217	-0.0030	272	1.1028	327	0.2176
53	1.4500	108	1.2672	163	-0.7075	218	0.6165	273	-0.2626	328	0.0516
54	-0.4815	109	-1.1675	164	-1.0753	219	-0.1882	274	-0.6537	329	1.0286
55	0.4842	110	0.8051	165	1.3903	220	-0.8508	275	-0.2321	330	0.5552

ตาราง 23 (ต่อ) แสดงค่าความสามารถของผู้สอบ (O) กรณีความสามารถของผู้สอบมีการแจกแจงแบบปกติ

คนที่	อ	คนที่	อ	คนที่	อ	คนที่	อ	คนที่	อ	คนที่	อ
331	-0.3822	386	0.8103	441	-2.5479	496	0.5463	551	0.5528	606	2.1535
332	-0.9145	387	-0.0739	442	0.9409	497	-0.6990	552	0.3588	607	0.3880
333	0.0641	388	-0.3938	443	-0.5696	498	-0.1683	553	0.1091	608	0.9881
334	-2.0354	389	0.9193	444	-0.5200	499	0.3585	554	0.5865	609	-0.2673
335	0.4019	390	-0.3005	445	1.3535	500	0.4541	555	-0.1884	610	-0.4692
336	1.6979	391	2.4989	446	-0.4370	501	2.8666	556	0.4203	611	-0.7852
337	-0.6978	392	0.1166	447	-0.2297	502	-0.3985	557	0.2834	612	0.9438
338	0.2337	393	2.3017	448	0.3837	503	0.2437	558	0.4186	613	-0.0774
339	0.8897	394	-0.3471	449	0.3129	504	0.2839	559	-1.3435	614	-0.1769
340	0.5665	395	0.1152	450	-0.4852	505	-0.0492	560	-1.3971	615	0.8353
341	1.0095	396	1.8678	451	0.3257	506	-1.1769	561	0.1685	616	-0.6662
342	0.4060	397	0.7484	452	1.1941	507	1.5114	562	-0.0009	617	1.5342
343	2.4223	398	0.1081	453	-0.4334	508	-1.3130	563	0.9907	618	-0.9982
344	0.8984	399	0.0305	454	0.4434	509	1.2794	564	-0.9575	619	0.2816
345	0.3619	400	0.1926	455	0.3940	510	-0.0616	565	0.6192	620	-0.9357
346	-0.4511	401	0.6107	456	0.7186	511	0.3434	566	0.7354	621	1.1048
347	0.3657	402	-0.1141	457	-1.9275	512	0.0437	567	1.6875	622	0.3114
348	0.3800	403	-0.4894	458	-1.4957	513	0.5328	568	2.9152	623	2.1558
349	-1.4926	404	-0.0467	459	-0.7907	514	0.4433	569	0.8030	624	0.5295
350	-1.2245	405	-1.6221	460	0.7769	515	-0.0097	570	0.6930	625	-1.7423
351	0.4061	406	0.8390	461	-0.9378	516	-2.0589	571	0.3280	626	0.9910
352	-0.4609	407	0.8870	462	-0.2266	517	-0.1617	572	-0.0240	627	0.4478
353	1.4155	408	1.7839	463	0.8374	518	0.2682	573	0.9659	628	1.0129
354	-0.1592	409	-1.4197	464	-1.7517	519	-1.8722	574	-0.8866	629	-0.0467
355	0.1757	410	-1.0580	465	-0.0587	520	0.1795	575	-0.3899	630	-1.3470
356	-0.1512	411	0.6537	466	1.8015	521	0.8110	576	-0.2240	631	-0.3123
357	0.8171	412	-1.8680	467	0.6395	522	-0.1531	577	-0.6158	632	1.3000
358	1.6891	413	-0.8968	468	0.4838	523	0.2234	578	1.0091	633	1.0886
359	-0.4738	414	0.1247	469	-1.1082	524	-0.9931	579	-0.5521	634	1.3477
360	-0.5641	415	0.5479	470	-1.8328	525	0.5407	580	-0.5028	635	0.4557
361	0.2998	416	-2.4928	471	-0.6143	526	0.9821	581	0.4843	636	0.0997
362	-0.5735	417	0.1488	472	-0.4174	527	-0.5978	582	0.2385	637	-0.8017
363	-1.1727	418	0.2766	473	-0.5058	528	1.7368	583	1.4768	638	0.6099
364	0.7825	419	0.4570	474	0.7986	529	0.8996	584	0.5534	639	1.2206
365	0.1228	420	-1.0216	475	0.7945	530	-0.2449	585	-0.6864	640	1.2867
366	0.1207	421	-0.9432	476	0.9445	531	-0.8723	586	0.5131	641	-1.1889
367	-1.4088	422	-0.5788	477	1.4132	532	0.0807	587	-0.7234	642	0.1530
368	-0.7940	423	-0.1015	478	0.6283	533	-1.3406	588	1.8704	643	0.1343
369	0.3703	424	-0.1966	479	-0.0394	534	0.6179	589	-0.4989	644	-0.0546
370	2.2133	425	-0.3172	480	0.8038	535	0.0833	590	-0.3761	645	-0.2244
371	1.0152	426	0.8229	481	-0.3424	536	0.4760	591	-0.1360	646	0.5396
372	2.1236	427	0.1682	482	1.2512	537	-0.8555	592	-0.1167	647	-1.8433
373	0.7144	428	0.8650	483	-0.6501	538	1.3273	593	-1.0052	648	1.7255
374	-0.6749	429	-0.1515	484	-1.1821	539	1.6150	594	0.1567	649	0.0445
375	-0.8831	430	-1.4812	485	-0.7267	540	-1.2686	595	-1.0369	650	-0.2109
376	-0.6442	431	-0.5344	486	1.3041	541	-0.1499	596	-2.3329	651	1.4999
377	-0.1590	432	-1.0367	487	0.2219	542	-1.3027	597	0.3398	652	0.6473
378	0.2742	433	-1.3299	488	-0.6177	543	-0.6499	598	0.1494	653	2.0526
379	-1.3504	434	0.2472	489	-0.8159	544	-0.1054	599	-0.6761	654	0.1204
380	-1.8773	435	0.5787	490	-0.0726	545	-0.7650	600	-0.4821	655	-0.2141
381	-1.8564	436	0.6148	491	-1.3199	546	0.2104	601	-0.2631	656	-0.5008
382	0.4464	437	1.8459	492	-0.2318	547	-0.3132	602	-0.8679	657	1.0299
383	0.7859	438	-0.2852	493	-1.0671	548	0.1506	603	0.9885	658	-1.2887
384	-2.2021	439	-0.1254	494	0.7237	549	1.3966	604	-0.3814	659	-0.3556
385	0.6242	440	0.2823	495	0.6416	550	-0.4841	605	-0.1305	660	-0.5594

ตาราง 23 (ต่อ) แสดงค่าความสามารถของผู้สอบ (Pi) กรณีความสามารถของผู้สอบมีการแจกแจงแบบปกติ

คนที่	Pi	คนที่	Pi	คนที่	Pi	คนที่	Pi	คนที่	Pi	คนที่	Pi
661	-0.4986	716	0.3890	771	0.2349	826	1.5955	881	0.0990	936	0.1991
662	-1.7366	717	-0.5645	772	1.6744	827	-0.2569	882	0.1784	937	0.9090
663	1.1485	718	-0.6779	773	-0.4043	828	1.3741	883	-0.0692	938	0.9701
664	0.1355	719	0.3071	774	-0.3782	829	0.5328	884	0.4566	939	0.0561
665	-0.4298	720	-2.2899	775	-2.2946	830	-0.3508	885	-0.0967	940	1.7214
666	0.1084	721	-0.3077	776	0.4235	831	-1.7126	886	0.8991	941	-0.3787
667	-0.5769	722	0.4045	777	-1.6644	832	0.3380	887	-0.3268	942	-1.2686
668	-0.1808	723	-0.8833	778	-0.5703	833	0.2922	888	1.5754	943	0.8637
669	0.9724	724	-1.4861	779	-1.0320	834	1.4095	889	-0.3930	944	-0.2241
670	-1.9498	725	-1.1370	780	-0.4720	835	-0.2477	890	-0.9618	945	0.3113
671	0.7307	726	-0.8255	781	-0.0325	836	-0.7048	891	-0.4062	946	-0.0579
672	1.5654	727	-0.1180	782	-0.7696	837	0.5282	892	-1.4586	947	0.6702
673	0.9241	728	-1.0842	783	0.4085	838	1.2582	893	-0.6482	948	-0.1706
674	-2.5855	729	1.0633	784	0.3808	839	-0.0431	894	-0.8882	949	-0.9623
675	-1.1676	730	-1.2254	785	0.1847	840	-0.5945	895	2.0183	950	-0.8542
676	-1.2917	731	1.8712	786	0.8013	841	-0.8344	896	1.1652	951	0.5717
677	-0.9999	732	-0.4202	787	1.0300	842	0.3500	897	0.5211	952	0.4105
678	0.0679	733	-0.4201	788	-0.3510	843	0.7693	898	-0.3884	953	0.2684
679	-0.6396	734	-0.8090	789	0.8195	844	-1.7613	899	0.3628	954	-1.6998
680	1.9443	735	-0.0737	790	-0.0037	845	-0.9607	900	-0.5273	955	-0.3504
681	0.7678	736	-2.7595	791	2.2642	846	-0.4573	901	-1.4990	956	-0.6531
682	0.2692	737	0.9216	792	-0.1318	847	1.2178	902	-0.6999	957	0.5621
683	-0.1980	738	0.8487	793	0.0325	848	-0.1513	903	0.0553	958	-0.9251
684	0.1537	739	-1.2840	794	0.0528	849	-1.1498	904	-0.5065	959	-0.7968
685	0.6761	740	0.0288	795	-0.6252	850	-2.8338	905	-0.7357	960	1.8145
686	0.1326	741	-0.9579	796	0.2434	851	0.1480	906	0.6367	961	-0.3681
687	-0.4168	742	-0.8071	797	2.2896	852	-2.5406	907	0.2520	962	-0.6094
688	0.6923	743	-0.7302	798	0.9070	853	0.6984	908	-0.8363	963	-0.1180
689	0.4472	744	-0.3139	799	1.1566	854	1.0069	909	0.9752	964	-0.5734
690	-0.8708	745	0.7333	800	1.3319	855	1.4233	910	-1.5066	965	0.4278
691	-0.6233	746	-0.4839	801	0.8405	856	0.3912	911	-1.1525	966	1.9123
692	0.4955	747	-1.2014	802	-1.0288	857	-1.1433	912	2.8034	967	-0.3916
693	-1.2253	748	1.3561	803	0.4262	858	-0.9026	913	0.4914	968	-1.1558
694	0.1529	749	-0.2275	804	1.3306	859	0.5308	914	-0.2176	969	0.3412
695	-1.1725	750	0.2530	805	-0.2193	860	0.1283	915	0.3040	970	-0.8401
696	-0.5822	751	0.2090	806	0.6602	861	1.1573	916	-0.1812	971	-0.2814
697	-0.5120	752	-1.5120	807	0.8781	862	-2.0239	917	0.4368	972	-0.9464
698	-1.3151	753	-0.8464	808	-0.3789	863	-1.5064	918	0.6706	973	0.7203
699	0.1458	754	0.1474	809	-0.2945	864	-0.3235	919	0.7960	974	0.1121
700	-1.5778	755	0.0653	810	-1.4762	865	1.8357	920	-1.0116	975	1.8218
701	0.7175	756	-1.8006	811	1.0644	866	1.5840	921	0.3575	976	-0.1578
702	0.3504	757	-1.9005	812	-0.2882	867	-0.0427	922	-2.5533	977	0.0191
703	1.8867	758	-1.3069	813	-0.5371	868	0.2096	923	-1.2367	978	1.1804
704	1.9010	759	-0.2727	814	1.6213	869	-0.5482	924	-1.4697	979	-0.2490
705	-0.9353	760	1.5271	815	-1.7492	870	-1.1652	925	0.5717	980	-0.2512
706	-1.8025	761	1.6402	816	-1.1914	871	-1.8921	926	-0.5738	981	-1.6922
707	-0.3599	762	0.1080	817	-0.7828	872	-0.0350	927	0.4587	982	-2.3172
708	-2.2962	763	-1.2902	818	-0.6385	873	-0.4697	928	-0.3299	983	-0.4182
709	-0.0410	764	0.6090	819	-1.7817	874	-1.4280	929	-0.3115	984	0.4236
710	-1.3090	765	1.3287	820	-0.2452	875	0.7559	930	0.9009	985	-0.1960
711	-1.6036	766	2.0722	821	-0.6737	876	0.3858	931	0.9787	986	-0.3163
712	0.6954	767	0.4325	822	1.0414	877	0.2246	932	-0.3818	987	-2.1330
713	0.0189	768	-0.8871	823	-0.9146	878	1.3501	933	0.3478	988	0.8678
714	-2.3247	769	0.4353	824	-1.2902	879	0.4973	934	1.4800	989	-0.3715
715	0.8541	770	0.7890	825	-1.1672	880	-1.0121	935	-0.0410	990	0.0528

ตาราง 23 (ต่อ) แสดงค่าความสามารถของผู้สอบ (O) กรณีความสามารถของผู้สอบมีการแจกแจงแบบปกติ

คนที่	อ	คนที่	อ	คนที่	อ	คนที่	อ
991	-0.9645	1046	-2.5913	1101	0.8420	1,156	0.7794
992	0.2351	1047	0.6273	1102	-1.6434	1,157	1.1903
993	0.2889	1048	-0.3820	1103	-0.5443	1,158	-0.4667
994	1.9000	1049	0.9370	1104	-0.0804	1,159	1.5083
995	-2.3484	1050	1.1112	1105	-0.6380	1,160	0.9970
996	-0.3019	1051	-0.4325	1106	-0.2561	1,161	-1.3165
997	1.9728	1052	0.6277	1107	1.0533	1,162	1.4784
998	0.3781	1053	-0.1345	1108	-0.3899	1,163	0.0765
999	0.8810	1054	0.4755	1109	0.8403	1,164	-0.2391
1000	-0.4920	1055	-0.1547	1110	0.2924	1,165	0.5790
1001	0.3518	1056	1.0155	1111	0.7374	1,166	-0.1216
1002	0.6539	1057	0.6346	1112	0.2137	1,167	-0.7763
1003	-0.1131	1058	-1.3341	1113	-0.3477	1,168	1.8605
1004	0.4512	1059	0.6716	1114	-1.2364	1,169	0.3072
1005	1.2730	1060	-0.0540	1115	0.2775	1,170	-1.0456
1006	1.3801	1061	-0.5131	1116	0.4957	1,171	0.0594
1007	-1.1072	1062	-1.7113	1117	-2.3314	1,172	1.7054
1008	-0.5483	1063	0.1965	1118	0.1482	1,173	-0.0239
1009	0.1652	1064	-0.8151	1119	0.6904	1,174	1.2846
1010	-0.0305	1065	-0.2353	1120	-0.1451	1,175	-0.3161
1011	-0.4692	1066	-1.6926	1121	-0.6173	1,176	0.1231
1012	0.3101	1067	-0.5353	1122	0.8518	1,177	-2.1019
1013	-0.3883	1068	0.7912	1123	0.4973	1,178	-1.6645
1014	-0.3065	1069	0.1237	1124	1.4625	1,179	-0.5219
1015	1.3058	1070	0.1442	1125	0.0904	1,180	-0.6355
1016	1.6664	1071	-1.4906	1126	0.9354	1,181	-0.7338
1017	-0.0655	1072	0.9589	1127	1.0554	1,182	0.3790
1018	-1.0021	1073	0.2902	1128	0.9450	1,183	-0.9941
1019	0.4285	1074	-0.8321	1129	-0.7211	1,184	-0.1693
1020	-1.1478	1075	-0.7711	1130	0.3595	1,185	1.8690
1021	-1.0560	1076	-0.2585	1131	0.0037	1,186	1.7782
1022	-0.7414	1077	-0.2498	1132	-0.3825	1,187	1.0254
1023	-0.6176	1078	0.7631	1133	-1.2352	1,188	-0.4213
1024	0.3024	1079	-1.2686	1134	-0.0815	1,189	0.5521
1025	0.5594	1080	0.8666	1135	0.4737	1,190	-1.2008
1026	-0.4091	1081	-0.5063	1136	-0.9192	1,191	0.9460
1027	0.3121	1082	-0.6211	1137	0.8698	1,192	-0.3371
1028	-1.9947	1083	-0.0927	1138	0.2079	1,193	-1.1825
1029	-0.5415	1084	-1.4474	1139	0.2921	1,194	-1.3871
1030	-1.6506	1085	-0.1001	1140	-0.2846	1,195	-0.7243
1031	0.7804	1086	-0.6282	1141	2.6197	1,196	-0.7075
1032	1.2316	1087	-0.2402	1142	-0.3279	1,197	0.0968
1033	0.7556	1088	0.8985	1143	0.9174	1,198	-0.8517
1034	-2.2929	1089	-0.4517	1144	1.3873	1,199	-1.7142
1035	-0.5881	1090	0.2228	1145	1.1250	1,200	-0.2901
1036	1.7171	1091	1.5983	1146	1.5327		
1037	-1.3356	1092	-0.7739	1147	0.4562		
1038	0.4383	1093	0.7380	1148	-0.8526		
1039	0.5245	1094	1.3902	1149	1.9792		
1040	-0.5633	1095	1.3559	1150	-0.4664		
1041	1.7983	1096	0.3763	1151	-1.6393		
1042	0.2283	1097	-1.0691	1152	-0.6848		
1043	-0.2647	1098	-0.2882	1153	0.0763		
1044	-0.4807	1099	1.7793	1154	-0.2604		
1045	-0.6230	1100	0.9342	1155	1.6143		

ตาราง 24 แสดงค่าความยากของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบปกติ

ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j
1	-0.5399	56	0.3833	111	-0.8447	166	1.1295	221	-0.7051	276	0.6092	331	-1.6868
2	0.9343	57	-0.5436	112	-0.4961	167	0.7472	222	0.8480	277	1.5021	332	1.4195
3	-0.5571	58	0.2610	113	0.3853	168	-1.8383	223	-0.2035	278	0.3337	333	-0.1634
4	1.0198	59	1.0112	114	0.0421	169	1.5539	224	0.8311	279	0.8204	334	0.7303
5	0.0643	60	0.8156	115	0.0623	170	1.2916	225	-0.0432	280	-0.7451	335	-1.0384
6	0.1148	61	-0.7793	116	0.2043	171	-1.0763	226	0.4331	281	-0.6632	336	0.5797
7	-0.1646	62	1.8207	117	0.1122	172	-0.2012	227	0.4400	282	1.0470	337	-0.2993
8	0.2878	63	1.1830	118	-0.5188	173	-0.8772	228	1.6743	283	-0.8660	338	-1.8313
9	0.9643	64	-0.0065	119	-0.5126	174	1.0642	229	-0.5406	284	0.6940	339	1.0626
10	-0.0957	65	0.5648	120	1.1817	175	0.0667	230	-0.0050	285	0.3105	340	-1.1287
11	0.1596	66	-1.2761	121	0.2032	176	-0.0787	231	0.7820	286	-0.8513	341	-0.1755
12	-0.1648	67	1.2771	122	-0.2099	177	0.5756	232	-0.9481	287	-1.7778	342	1.2475
13	0.4444	68	0.8701	123	-0.8067	178	0.7672	233	0.0873	288	0.2062	343	0.2686
14	-0.0655	69	0.8149	124	0.4131	179	1.1668	234	0.0879	289	-0.5509	344	0.6088
15	-0.0334	70	1.1227	125	-0.6054	180	0.9806	235	-0.4811	290	-1.1858	345	-1.4992
16	-0.2773	71	0.4133	126	-1.0823	181	0.4846	236	0.4047	291	1.4601	346	-0.2601
17	1.0078	72	1.3199	127	-0.1174	182	-1.0353	237	0.3489	292	0.1147	347	-0.4995
18	1.8518	73	1.2574	128	0.8447	183	1.1234	238	0.2516	293	-1.0275	348	0.5400
19	-0.4602	74	1.0863	129	0.8929	184	0.1009	239	0.2744	294	1.3864	349	0.2293
20	-1.6973	75	-0.7587	130	1.1078	185	-0.9018	240	0.1384	295	1.2139	350	0.6261
21	0.2790	76	1.6194	131	-1.6279	186	-0.9967	241	-0.5397	296	0.7350	351	-0.7427
22	-0.1915	77	0.2240	132	-0.8133	187	-0.3848	242	-0.1891	297	-0.3917	352	0.6000
23	0.4634	78	-0.7008	133	-0.8174	188	-0.0185	243	-0.5444	298	0.4892	353	-1.8537
24	-0.9350	79	0.2492	134	-0.8688	189	-1.5585	244	-0.4025	299	0.6898	354	-1.3349
25	0.6733	80	-0.4991	135	-0.4366	190	-0.0370	245	1.1596	300	1.6121	355	0.0341
26	-1.0293	81	-0.0612	136	-0.7459	191	-1.3204	246	-0.2504	301	-0.3344	356	-0.0742
27	0.0189	82	1.2371	137	-0.1711	192	0.7285	247	1.7805	302	-0.5070	357	-1.1372
28	1.2157	83	0.1849	138	0.4419	193	-0.8806	248	-0.2191	303	-0.9703	358	0.3120
29	-0.0952	84	-1.0116	139	0.7721	194	-1.6558	249	-0.1888	304	-0.3948	359	-0.4655
30	1.2554	85	-0.9768	140	0.9267	195	1.9754	250	0.0721	305	0.6644	360	-1.5488
31	-1.8075	86	0.8348	141	-0.1699	196	-0.8942	251	-1.8080	306	0.8439	361	0.1699
32	-1.5464	87	0.3101	142	1.5002	197	-0.7289	252	0.3222	307	1.0585	362	-0.1479
33	0.6090	88	0.9270	143	0.5824	198	0.5672	253	1.4312	308	-1.0278	363	-0.1045
34	1.0565	89	-0.7634	144	-1.8841	199	-0.0503	254	-0.9414	309	-1.8796	364	-0.8981
35	0.9868	90	0.3866	145	-0.7813	200	0.2666	255	0.9780	310	1.1809	365	-0.5468
36	0.0651	91	1.5333	146	0.3964	201	0.8254	256	-0.9949	311	1.3119	366	-1.0231
37	-1.1806	92	1.4798	147	-0.8820	202	0.3847	257	0.1497	312	0.9292	367	-1.0826
38	-0.0590	93	-0.5479	148	1.6774	203	0.4570	258	-0.1000	313	0.1129	368	-0.8697
39	-1.2126	94	-1.3342	149	1.1319	204	-1.2395	259	-0.4582	314	-1.0549	369	0.5495
40	0.2546	95	0.2490	150	-0.8227	205	-1.9257	260	0.3822	315	-1.8327	370	-0.7372
41	0.4635	96	1.6885	151	-1.0168	206	-1.7797	261	-0.9445	316	-1.9763	371	0.6259
42	-0.1142	97	1.8244	152	1.0350	207	-0.5990	262	1.0048	317	-0.5578	372	0.8686
43	-0.4216	98	1.6641	153	-0.4282	208	-0.4556	263	-0.6883	318	-0.0112	373	0.9953
44	0.0939	99	1.1672	154	-0.3004	209	1.2086	264	0.5398	319	-0.3330	374	-0.5369
45	1.2578	100	0.1885	155	0.0250	210	0.1175	265	-0.8039	320	-1.5538	375	-0.8309
46	-0.2327	101	-0.6759	156	-0.3306	211	0.5121	266	1.3542	321	-1.3208	376	-1.1775
47	-1.1807	102	-1.9123	157	1.0764	212	0.8589	267	1.0781	322	0.8733	377	0.6969
48	1.4424	103	-0.1222	158	0.7775	213	0.8640	268	-1.6887	323	1.5288	378	-0.5982
49	-1.0598	104	-1.0805	159	-0.8166	214	-0.1071	269	-0.0468	324	0.6984	379	-0.7773
50	1.0766	105	-0.8514	160	1.3059	215	1.4412	270	1.0403	325	-0.6949	380	0.4952
51	0.3788	106	-0.4483	161	0.0812	216	1.0128	271	-0.0442	326	-0.1273	381	-0.2220
52	-0.2060	107	-0.2057	162	-0.2210	217	-0.3670	272	1.7164	327	0.1383	382	0.8825
53	0.5345	108	-0.3616	163	-0.7113	218	-0.3223	273	1.0985	328	-1.5038	383	-0.3080
54	-1.8785	109	0.8622	164	1.4041	219	0.5669	274	0.0963	329	0.6174	384	1.2330
55	0.6314	110	-0.0014	165	-0.5690	220	-0.2332	275	-1.4956	330	1.7958	385	-1.1186

ตาราง 24 (ต่อ) แสดงค่าความยากของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบปกติ

ข้อที่	β_j	ข้อที่	β_j
386	-0.3223	441	-0.8004
387	-0.5436	442	-0.2048
388	-0.9656	443	0.9979
389	-0.3875	444	-1.0013
390	-0.8911	445	-0.2328
391	-0.4209	446	1.0373
392	0.1227	447	1.2380
393	-0.5687	448	0.3972
394	0.0157	449	1.6104
395	-0.9686	450	0.6710
396	1.3505	451	1.5321
397	0.7824	452	-1.1315
398	-0.5034	453	-0.4533
399	-0.3832	454	0.5630
400	0.4558	455	-0.7526
401	-0.6824	456	0.4638
402	-0.8550	457	-0.4752
403	-0.9297	458	-1.4367
404	-0.4892	459	0.3289
405	0.4617	460	-0.0537
406	0.2993	461	0.3273
407	0.6800	462	0.6433
408	1.2904	463	1.6206
409	-0.9705	464	0.0632
410	0.2030	465	0.1218
411	-0.4428	466	0.3696
412	-0.9599	467	0.1632
413	0.2670	468	-0.2087
414	0.6713	469	-0.6464
415	-0.1068	470	-0.1078
416	-0.8592	471	1.3644
417	-0.1520	472	-0.5636
418	-0.8242	473	-1.7125
419	-0.6927	474	-0.1830
420	0.0463	475	0.3863
421	0.8954	476	0.3881
422	-0.3872	477	-0.5435
423	1.0584	478	-0.7912
424	-0.2574	479	-0.7142
425	1.1522	480	0.5868
426	-1.6351		
427	0.6682		
428	0.5124		
429	-1.8040		
430	0.2226		
431	0.9287		
432	-0.0877		
433	0.8124		
434	0.4862		
435	-1.7643		
436	0.2693		
437	0.2468		
438	0.2331		
439	-0.8613		
440	0.7099		

ตาราง 25 แสดงค่าความยากของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบปกติ

ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j
1	1.7640	56	-0.3255	111	1.5939	166	1.9164	221	1.6192	276	1.1805	331	1.2924
2	1.6297	57	1.7128	112	1.8780	167	1.9733	222	1.6429	277	1.3132	332	0.0781
3	-0.3625	58	0.9367	113	1.8286	168	1.5657	223	1.4313	278	1.3981	333	1.5788
4	-0.7182	59	1.0014	114	1.9374	169	1.9620	224	-0.2413	279	1.0036	334	1.1908
5	1.6271	60	0.5634	115	1.1526	170	1.1837	225	0.5910	280	1.3885	335	0.7862
6	1.5814	61	1.0897	116	0.9292	171	1.5534	226	1.6617	281	1.1943	336	1.8680
7	1.6670	62	1.4244	117	1.1754	172	1.2626	227	-0.8627	282	1.8745	337	-0.0861
8	0.4438	63	0.4467	118	0.7780	173	1.1899	228	-0.7749	283	1.4053	338	0.9644
9	1.9543	64	0.9027	119	1.0829	174	0.8134	229	0.8256	284	1.5189	339	1.9335
10	1.7912	65	1.3566	120	1.8191	175	0.2828	230	0.3986	285	0.9960	340	-0.6263
11	0.7789	66	1.9214	121	-0.4234	176	0.3032	231	-0.2334	286	1.3386	341	0.7867
12	1.5721	67	1.3421	122	1.5294	177	0.0917	232	1.3446	287	0.8810	342	1.1925
13	1.5129	68	1.9724	123	-0.6730	178	0.5408	233	1.4162	288	1.9778	343	1.5518
14	0.0605	69	1.1371	124	0.5274	179	0.2054	234	-0.3702	289	1.3967	344	1.8414
15	1.9177	70	-0.0042	125	1.7752	180	1.1979	235	1.9593	290	1.8097	345	0.7905
16	1.7074	71	0.5642	126	0.6037	181	1.7928	236	-0.2474	291	0.5837	346	1.8299
17	1.5929	72	1.0075	127	0.2283	182	1.7038	237	1.3195	292	0.5300	347	0.3883
18	1.1141	73	0.8115	128	1.1978	183	0.3961	238	0.4971	293	1.4904	348	0.5700
19	1.5932	74	0.7229	129	1.4495	184	-0.2101	239	1.8369	294	1.9369	349	0.9574
20	0.3118	75	1.5782	130	1.7807	185	1.9168	240	1.4255	295	0.3840	350	1.3814
21	0.9957	76	1.9843	131	1.4147	186	1.6065	241	1.0089	296	1.3620	351	-0.3888
22	0.0070	77	1.8561	132	0.8854	187	-0.2863	242	0.9930	297	-0.1770	352	0.7352
23	1.9701	78	0.8544	133	1.6776	188	1.6821	243	-0.5336	298	1.6683	353	0.1315
24	1.3987	79	-0.8457	134	1.4927	189	1.9504	244	-0.5352	299	1.0674	354	0.7214
25	1.2911	80	1.8188	135	1.9295	190	1.7018	245	0.9276	300	1.9794	355	1.1399
26	-0.4325	81	1.4157	136	1.3185	191	1.1185	246	0.6111	301	1.9859	356	1.4347
27	1.8637	82	1.0406	137	0.6726	192	1.6941	247	1.7723	302	0.7978	357	1.0708
28	1.9266	83	0.5535	138	1.3175	193	1.0765	248	0.8501	303	0.5142	358	1.7937
29	1.2252	84	0.4123	139	1.4682	194	1.5143	249	1.7125	304	1.1143	359	1.7676
30	0.9920	85	1.7644	140	0.4237	195	-0.6658	250	0.9841	305	0.8764	360	0.4707
31	-0.1942	86	0.9884	141	1.6611	196	1.1547	251	1.7095	306	1.9433	361	1.8957
32	0.0836	87	1.7115	142	1.3929	197	1.9436	252	0.6741	307	0.2769	362	1.9911
33	-0.3727	88	0.1042	143	1.3509	198	1.5362	253	1.7421	308	0.8253	363	1.8340
34	1.7460	89	1.2583	144	1.2589	199	1.7116	254	0.6510	309	1.6397	364	1.1270
35	1.3067	90	1.8572	145	1.7094	200	0.9429	255	0.8522	310	1.8423	365	1.9385
36	1.0170	91	0.6668	146	-0.2504	201	1.1104	256	1.2796	311	1.6315	366	1.7542
37	0.6729	92	0.9645	147	1.8838	202	0.9053	257	0.5987	312	1.2042	367	1.2489
38	0.1666	93	1.9484	148	0.1874	203	1.1415	258	0.1376	313	1.8883	368	0.6393
39	0.9471	94	1.8980	149	0.5340	204	1.2636	259	1.3626	314	-0.6898	369	0.7191
40	0.2146	95	1.3935	150	1.5949	205	1.4560	260	-0.5764	315	1.0918	370	1.9832
41	0.3190	96	1.9608	151	1.5172	206	-0.2917	261	1.1874	316	1.6668	371	1.4933
42	0.9636	97	0.1405	152	1.0992	207	1.2976	262	0.8528	317	1.2802	372	1.0410
43	0.3137	98	-0.6547	153	1.7397	208	1.6282	263	1.3572	318	1.3614	373	1.5995
44	1.2217	99	1.8089	154	0.4279	209	1.5580	264	1.7968	319	1.5287	374	0.9852
45	1.6095	100	0.7784	155	1.0271	210	0.7436	265	0.0528	320	1.4900	375	1.4432
46	0.9208	101	1.2180	156	1.6051	211	-0.0330	266	0.4561	321	1.9006	376	1.9690
47	0.6547	102	0.7347	157	0.1282	212	1.9393	267	1.4095	322	1.8512	377	1.7095
48	1.7069	103	0.8487	158	1.0040	213	1.4528	268	0.7894	323	1.8619	378	1.3223
49	1.8506	104	0.4661	159	1.8310	214	1.3543	269	0.7050	324	0.3384	379	0.6298
50	1.0603	105	0.5088	160	1.6796	215	1.8270	270	-0.2288	325	0.0082	380	0.6244
51	-0.0071	106	1.6387	161	1.4201	216	1.1616	271	1.5088	326	0.2291	381	1.6158
52	0.9897	107	1.3001	162	1.5352	217	1.6799	272	1.3294	327	1.7808	382	1.6705
53	1.2437	108	0.0474	163	1.7593	218	1.5245	273	1.5709	328	0.0403	383	1.9163
54	1.9715	109	1.3657	164	0.8833	219	1.4036	274	1.4894	329	1.9769	384	1.8934
55	1.8627	110	1.7820	165	0.0070	220	0.0996	275	1.8600	330	0.1267	385	0.2985

ตาราง 25 (ต่อ) แสดงค่าความยาวของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบปกติ

ข้อที่	β_j	ข้อที่	β_j
386	1.4693	441	1.8281
387	0.8414	442	1.9329
388	0.5575	443	0.9952
389	1.8675	444	1.5002
390	0.5753	445	1.8262
391	0.7461	446	0.9866
392	-0.5330	447	0.4590
393	1.8262	448	0.7534
394	0.1153	449	0.4045
395	1.2223	450	1.9536
396	1.9768	451	1.5664
397	1.2802	452	-0.0252
398	1.9590	453	1.9902
399	-0.3503	454	0.7982
400	1.1832	455	1.6361
401	0.1736	456	1.0480
402	1.6604	457	0.8403
403	-0.2825	458	1.0416
404	1.7206	459	0.8619
405	1.8871	460	0.7885
406	0.8303	461	-0.0576
407	1.6593	462	1.8779
408	1.3203	463	1.2386
409	-0.1811	464	-0.7410
410	1.3431	465	1.8655
411	1.7579	466	1.1576
412	1.1583	467	1.5863
413	1.5410	468	0.0015
414	1.1286	469	0.6850
415	1.0465	470	1.9088
416	0.9529	471	0.9918
417	0.4328	472	1.7665
418	1.8252	473	1.0042
419	0.7521	474	1.2573
420	1.7322	475	1.4105
421	1.4176	476	0.4400
422	0.8343	477	1.8224
423	0.6282	478	1.6004
424	1.9648	479	0.5403
425	0.9744	480	0.4957
426	0.0140		
427	-0.1765		
428	1.3584		
429	0.4089		
430	0.0768		
431	1.2817		
432	0.4670		
433	0.0310		
434	1.8035		
435	1.6474		
436	1.0649		
437	1.1473		
438	1.8544		
439	0.5509		
440	1.4061		

ตาราง 26 แสดงค่าความยากของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบเบ้ขวา

ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j	ข้อที่	β_j
1	-1.5843	56	-1.4510	111	-1.8006	166	0.1302	221	-1.3628	276	-1.4739	331	-1.8698
2	-1.0965	57	-1.7342	112	-1.4916	167	-1.9572	222	-0.4960	277	-1.3142	332	-1.9843
3	1.2106	58	-0.4963	113	0.6888	168	-1.1783	223	0.9715	278	-1.5942	333	-1.7390
4	-1.8642	59	-0.6429	114	-1.7138	169	-1.8430	224	-1.5675	279	-0.7716	334	-0.3211
5	-0.8374	60	-0.3561	115	0.5289	170	-1.7548	225	-1.2532	280	-1.7577	335	-1.3473
6	-1.1956	61	-1.0714	116	-1.4396	171	-1.0882	226	-1.2304	281	-1.0464	336	-1.6900
7	0.2107	62	-1.4580	117	-1.9356	172	-1.8865	227	-1.0559	282	-0.7851	337	-1.6756
8	-1.9787	63	-1.0189	118	-0.9235	173	-0.3045	228	-0.1176	283	-1.8714	338	-1.7785
9	-1.8919	64	-1.7473	119	-1.3891	174	-0.1713	229	-0.7889	284	-1.6711	339	-1.4725
10	-1.0251	65	-1.9339	120	-1.4617	175	-0.3348	230	-1.8702	285	-1.6905	340	0.5227
11	-1.2505	66	-1.3403	121	-1.6597	176	-1.7759	231	-1.2201	286	-1.3679	341	-0.9173
12	0.9161	67	0.7837	122	-0.6101	177	-0.8871	232	0.3743	287	-0.7006	342	-0.8094
13	-1.2387	68	-1.9588	123	0.5775	178	-1.7649	233	-0.9556	288	-1.2892	343	-1.9847
14	0.1858	69	-1.0022	124	-0.6049	179	-1.5342	234	-1.2431	289	-0.8220	344	-1.6870
15	-1.1961	70	-1.1679	125	-1.2950	180	-1.7866	235	-1.6623	290	-1.6965	345	-0.7084
16	-1.3363	71	0.2247	126	-1.6315	181	-1.7957	236	-1.8829	291	-1.6919	346	-0.4876
17	-1.4035	72	-1.2956	127	-1.9276	182	-0.9905	237	-1.0316	292	-1.0680	347	0.5506
18	0.6624	73	0.2720	128	-1.5606	183	-1.0203	238	-1.2927	293	-1.6483	348	-1.4120
19	-1.3929	74	-0.1955	129	-1.3800	184	-0.5716	239	-1.9145	294	0.1663	349	-1.5832
20	-0.2163	75	-1.0656	130	-1.0836	185	0.7576	240	-1.8972	295	-1.5163	350	-1.3683
21	-1.5965	76	-0.0045	131	-0.8827	186	0.4073	241	-0.6469	296	-1.4530	351	-1.7893
22	-0.3584	77	-0.4548	132	0.0442	187	-1.8506	242	-1.8541	297	-1.9371	352	-0.2716
23	-1.4466	78	-1.8199	133	-1.7234	188	-1.9906	243	0.3788	298	-0.3833	353	-1.6220
24	0.7757	79	-0.9514	134	-0.4643	189	-0.3926	244	-0.1230	299	-1.6188	354	-0.2124
25	-1.1971	80	-0.3743	135	-1.4725	190	-1.4953	245	-1.8402	300	0.2879	355	-0.0074
26	-1.2256	81	-1.9835	136	-0.1048	191	-1.8903	246	-1.2150	301	-1.0767	356	-1.9253
27	-1.5370	82	-1.2916	137	-1.4408	192	-0.9984	247	0.2765	302	-0.7781	357	-1.3775
28	-1.3659	83	-1.9644	138	-0.5624	193	-1.9719	248	-0.2539	303	-1.4747	358	-1.8831
29	-1.4676	84	-1.7256	139	-1.3469	194	-1.3947	249	-0.1911	304	-1.4074	359	-1.7473
30	-0.1768	85	-0.9667	140	-0.5933	195	-1.0258	250	1.1819	305	-1.0378	360	-0.9437
31	-1.2258	86	-1.7619	141	-1.1322	196	-1.1786	251	-0.8293	306	-0.1053	361	-0.6771
32	-1.7867	87	-0.2712	142	-1.2863	197	-1.6333	252	-0.0869	307	-1.6437	362	-1.5656
33	0.1422	88	-1.1859	143	-1.7909	198	-1.7366	253	0.3337	308	-1.7957	363	-1.4131
34	-1.3295	89	-1.9935	144	-0.7880	199	-1.3457	254	0.2894	309	-0.8224	364	-1.0954
35	-1.6258	90	-0.3318	145	-1.0563	200	-1.8243	255	-1.8109	310	-1.6872	365	1.3657
36	-0.3733	91	0.6178	146	-1.5625	201	-0.8864	256	-1.9581	311	-1.8129	366	-0.8905
37	0.2138	92	-0.3743	147	-0.4342	202	-0.7966	257	-1.8525	312	-0.0142	367	-1.9303
38	-0.8489	93	-1.9024	148	-1.5047	203	0.7184	258	-0.0243	313	-1.1159	368	-1.7780
39	-1.8370	94	-1.6454	149	-0.1650	204	-1.7295	259	-1.1372	314	-1.9932	369	-1.1805
40	-0.4415	95	-1.3668	150	-1.4015	205	0.6078	260	-1.2552	315	-1.9274	370	0.9896
41	-1.9427	96	-0.6756	151	-1.1482	206	-1.9493	261	-0.9260	316	-0.8565	371	-0.8100
42	-1.6558	97	-0.7982	152	-1.1173	207	-0.9590	262	-1.7203	317	-1.9224	372	0.9465
43	-1.7590	98	-0.0062	153	-0.8292	208	-1.6748	263	-1.1205	318	-1.2274	373	-0.2229
44	-1.5072	99	-1.3273	154	-1.0095	209	-0.1636	264	-1.8149	319	-1.9413	374	-1.5084
45	0.5207	100	0.3959	155	-0.9956	210	-1.3604	265	0.4449	320	-1.7492	375	-1.7935
46	-1.4334	101	-0.5068	156	-1.2063	211	-1.8698	266	-0.0798	321	-1.5214	376	-0.8167
47	-1.1752	102	-1.8000	157	-1.4618	212	-1.2779	267	-1.1319	322	-1.0696	377	-0.8382
48	-1.9270	103	-1.8717	158	-1.6157	213	-1.5979	268	-1.6959	323	-1.8654	378	-1.7842
49	-1.9863	104	0.5068	159	-1.9549	214	-0.7972	269	-0.7574	324	-1.9946	379	-1.0625
50	-1.2098	105	0.7803	160	1.1717	215	-1.6040	270	-1.1225	325	-0.2055	380	-1.4174
51	-1.2547	106	-0.4923	161	-1.2908	216	0.2337	271	0.0870	326	-0.1264	381	0.0816
52	-1.5838	107	0.2784	162	-1.1776	217	-1.6831	272	-1.9506	327	-1.4183	382	-1.3974
53	-1.3999	108	-1.4160	163	0.4667	218	0.2989	273	-1.9162	328	-1.6692	383	-0.5134
54	-1.4728	109	-1.8620	164	-1.3049	219	-1.6672	274	-0.4106	329	-0.2814	384	-0.8978
55	-1.8097	110	-0.6927	165	-0.9576	220	-0.7920	275	-1.5409	330	-0.8851	385	1.3449

ตาราง 26 (ต่อ) แสดงค่าความยากของข้อสอบ (β_j) กรณีค่าความยากของข้อสอบมีการแจกแจงแบบปกติ

ข้อที่	β_j	ข้อที่	β_j
386	0.2223	441	1.0744
387	1.1508	442	-0.3323
388	-0.9191	443	0.1107
389	-1.9637	444	-1.7336
390	-0.4172	445	-0.9891
391	-0.5042	446	-0.1208
392	-1.7484	447	-1.7606
393	-1.8617	448	-0.4206
394	-1.2103	449	-1.9269
395	-1.5142	450	-0.3604
396	0.6777	451	-1.5898
397	-1.6558	452	-1.5638
398	-0.1851	453	-1.4640
399	-0.9845	454	-1.5677
400	1.0325	455	-0.0470
401	-1.7626	456	0.2067
402	-0.1841	457	-1.6855
403	1.4067	458	-1.6747
404	-1.8729	459	0.2453
405	-0.9114	460	-0.7476
406	-1.1344	461	-1.5721
407	-1.9445	462	-1.7180
408	-1.4140	463	-1.3122
409	-0.6811	464	-1.5610
410	-1.1518	465	-1.5834
411	-1.9684	466	-0.0169
412	-1.6797	467	-0.9436
413	-1.5964	468	-0.5583
414	-0.2574	469	-1.8829
415	-0.3661	470	-1.7362
416	1.0700	471	-1.6106
417	-1.5983	472	-1.4953
418	-1.2491	473	0.1296
419	-1.0874	474	-1.7758
420	-1.2936	475	-0.2500
421	-1.5807	476	-0.7239
422	-1.9231	477	-1.1468
423	-0.2419	478	-1.3513
424	-1.6803	479	-1.7307
425	-1.1413	480	0.6442
426	-1.8441		
427	-1.0808		
428	-0.4808		
429	-1.8773		
430	-1.7359		
431	-1.9228		
432	-1.3081		
433	0.3048		
434	-0.0430		
435	-1.9769		
436	-1.2936		
437	-1.2512		
438	-0.3801		
439	-1.0075		
440	-0.0924		

ตาราง 27 แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่มแบบเมตริกซ์เชิงพหุคูณ

กรณีข้อสอบมีลักษณะการแจกแจงค่าความยาวของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบ 600 คน

No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
1	0.49163	0.05620	0.04790	0.14588	0.24192	57	0.49157	0.05648	0.04760	0.14543	0.24154
2	0.49151	0.05654	0.04750	0.14558	0.24168	58	0.49243	0.05632	0.04759	0.14588	0.24192
3	0.49157	0.05602	0.04771	0.14621	0.24221	59	0.49101	0.05603	0.04775	0.14564	0.24173
4	0.49146	0.05596	0.04808	0.14602	0.24202	60	0.49159	0.05595	0.04821	0.14583	0.24187
5	0.49038	0.05636	0.04764	0.14576	0.24182	61	0.49262	0.05595	0.04766	0.14617	0.24216
6	0.49195	0.05581	0.04841	0.14583	0.24187	62	0.49189	0.05605	0.04836	0.14567	0.24173
7	0.49132	0.05560	0.04860	0.14617	0.24216	63	0.49150	0.05631	0.04772	0.14590	0.24192
8	0.49194	0.05648	0.04742	0.14609	0.24211	64	0.49109	0.05588	0.04810	0.14608	0.24206
9	0.49123	0.05573	0.04792	0.14636	0.24230	65	0.49089	0.05604	0.04776	0.14615	0.24216
10	0.49187	0.05572	0.04796	0.14616	0.24216	66	0.49275	0.05604	0.04810	0.14592	0.24197
11	0.49176	0.05587	0.04804	0.14608	0.24206	67	0.49208	0.05625	0.04804	0.14579	0.24182
12	0.49190	0.05605	0.04766	0.14580	0.24187	68	0.49187	0.05570	0.04810	0.14608	0.24206
13	0.49117	0.05615	0.04782	0.14596	0.24197	69	0.49120	0.05627	0.04771	0.14590	0.24192
14	0.49269	0.05589	0.04744	0.14583	0.24187	70	0.49180	0.05610	0.04803	0.14592	0.24197
15	0.49125	0.05597	0.04792	0.14609	0.24206	71	0.49133	0.05611	0.04807	0.14583	0.24187
16	0.49203	0.05618	0.04791	0.14596	0.24197	72	0.49197	0.05632	0.04784	0.14591	0.24192
17	0.49176	0.05650	0.04726	0.14587	0.24192	73	0.49196	0.05592	0.04785	0.14603	0.24202
18	0.49121	0.05620	0.04805	0.14583	0.24187	74	0.49174	0.05582	0.04803	0.14615	0.24216
19	0.49136	0.05626	0.04819	0.14580	0.24168	75	0.49120	0.05616	0.04752	0.14610	0.24211
20	0.49182	0.05562	0.04788	0.14582	0.24187	76	0.49143	0.05608	0.04777	0.14606	0.24206
21	0.49126	0.05632	0.04792	0.14576	0.24182	77	0.49217	0.05605	0.04795	0.14609	0.24206
22	0.49176	0.05597	0.04788	0.14557	0.24168	78	0.49102	0.05619	0.04772	0.14607	0.24206
23	0.49154	0.05587	0.04801	0.14611	0.24211	79	0.49192	0.05588	0.04820	0.14596	0.24197
24	0.49175	0.05626	0.04784	0.14590	0.24192	80	0.49151	0.05608	0.04791	0.14607	0.24206
25	0.49132	0.05601	0.04836	0.14596	0.24197	81	0.49201	0.05577	0.04789	0.14583	0.24187
26	0.49233	0.05633	0.04738	0.14635	0.24230	82	0.49121	0.05578	0.04793	0.14620	0.24216
27	0.49205	0.05596	0.04818	0.14592	0.24197	83	0.49167	0.05627	0.04790	0.14592	0.24192
28	0.49047	0.05586	0.04746	0.14626	0.24221	84	0.49129	0.05608	0.04776	0.14619	0.24216
29	0.49136	0.05623	0.04788	0.14587	0.24192	85	0.49205	0.05642	0.04792	0.14560	0.24168
30	0.49190	0.05611	0.04816	0.14583	0.24173	86	0.49281	0.05603	0.04755	0.14645	0.24240
31	0.49206	0.05620	0.04796	0.14583	0.24187	87	0.49148	0.05636	0.04785	0.14586	0.24187
32	0.49155	0.05624	0.04784	0.14599	0.24202	88	0.49156	0.05606	0.04774	0.14626	0.24221
33	0.49098	0.05624	0.04775	0.14605	0.24206	89	0.49180	0.05599	0.04788	0.14617	0.24216
34	0.49153	0.05617	0.04803	0.14587	0.24192	90	0.49149	0.05608	0.04805	0.14583	0.24187
35	0.49139	0.05633	0.04752	0.14560	0.24168	91	0.49078	0.05594	0.04783	0.14627	0.24226
36	0.49217	0.05551	0.04760	0.14614	0.24211	92	0.49197	0.05627	0.04808	0.14578	0.24178
37	0.49077	0.05586	0.04784	0.14633	0.24230	93	0.49096	0.05586	0.04791	0.14608	0.24202
38	0.49086	0.05597	0.04758	0.14579	0.24182	94	0.49122	0.05649	0.04778	0.14580	0.24182
39	0.49084	0.05609	0.04792	0.14601	0.24202	95	0.49189	0.05522	0.04780	0.14585	0.24187
40	0.49138	0.05611	0.04763	0.14608	0.24206	96	0.49117	0.05616	0.04784	0.14602	0.24202
41	0.49165	0.05632	0.04768	0.14576	0.24182	97	0.49163	0.05625	0.04763	0.14610	0.24211
42	0.49107	0.05615	0.04784	0.14585	0.24187	98	0.49173	0.05593	0.04794	0.14620	0.24216
43	0.49158	0.05624	0.04751	0.14602	0.24202	99	0.49234	0.05636	0.04794	0.14574	0.24182
44	0.49238	0.05598	0.04798	0.14608	0.24206	100	0.49136	0.05611	0.04809	0.14557	0.24163
45	0.49200	0.05634	0.04789	0.14574	0.24178	101	0.49195	0.05612	0.04684	0.14612	0.24211
46	0.49141	0.05638	0.04796	0.14571	0.24178	102	0.49213	0.05627	0.04824	0.14547	0.24158
47	0.49157	0.05622	0.04790	0.14596	0.24197	103	0.49162	0.05652	0.04745	0.14610	0.24211
48	0.49109	0.05591	0.04822	0.14583	0.24187	104	0.49198	0.05630	0.04802	0.14576	0.24182
49	0.49185	0.05583	0.04817	0.14605	0.24206	105	0.49105	0.05572	0.04826	0.14606	0.24206
50	0.49096	0.05581	0.04744	0.14594	0.24197	106	0.49224	0.05592	0.04797	0.14615	0.24216
51	0.49150	0.05571	0.04815	0.14604	0.24206	107	0.49225	0.05631	0.04781	0.14584	0.24187
52	0.49130	0.05607	0.04817	0.14584	0.24187	108	0.49203	0.05613	0.04815	0.14546	0.24158
53	0.49191	0.05641	0.04713	0.14576	0.24182	109	0.49157	0.05609	0.04769	0.14630	0.24226
54	0.49140	0.05620	0.04781	0.14563	0.24187	110	0.49123	0.05598	0.04831	0.14578	0.24182
55	0.49222	0.05621	0.04734	0.14618	0.24216	111	0.49084	0.05610	0.04804	0.14589	0.24192
56	0.49184	0.05628	0.04794	0.14577	0.24182	112	0.49074	0.05575	0.04800	0.14624	0.24221

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่มแบบตวิภาคีเชิงพหุคูณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน											
No.	Mean	VarE	VarI	VarEl	S.E.Mean	No.	Mean	VarE	VarI	VarEl	S.E.Mean
113	0.49217	0.05692	0.04794	0.14578	0.24182	169	0.49186	0.05604	0.04804	0.14588	0.24192
114	0.49211	0.05647	0.04673	0.14613	0.24211	170	0.49187	0.05618	0.04810	0.14580	0.24187
115	0.49116	0.05567	0.04839	0.14575	0.24182	171	0.49186	0.05598	0.04828	0.14564	0.24173
116	0.49160	0.05557	0.04825	0.14602	0.24202	172	0.49115	0.05621	0.04770	0.14607	0.24206
117	0.49065	0.05655	0.04755	0.14564	0.24173	173	0.49234	0.05655	0.04762	0.14591	0.24192
118	0.49194	0.05801	0.04787	0.14803	0.24202	174	0.49167	0.05618	0.04772	0.14594	0.24197
119	0.49094	0.05607	0.04767	0.14630	0.24226	175	0.49178	0.05652	0.04758	0.14568	0.24173
120	0.49220	0.05602	0.04815	0.14548	0.24158	176	0.49153	0.05620	0.04759	0.14603	0.24202
121	0.49151	0.05610	0.04793	0.14595	0.24197	177	0.49148	0.05641	0.04784	0.14579	0.24182
122	0.49176	0.05604	0.04778	0.14624	0.24221	178	0.49105	0.05637	0.04793	0.14571	0.24178
123	0.49171	0.05600	0.04755	0.14612	0.24211	179	0.49186	0.05606	0.04812	0.14589	0.24192
124	0.49182	0.05530	0.04807	0.14605	0.24206	180	0.49058	0.05595	0.04801	0.14598	0.24202
125	0.49207	0.05622	0.04775	0.14606	0.24206	181	0.49147	0.05556	0.04839	0.14572	0.24176
126	0.49193	0.05612	0.04812	0.14581	0.24187	182	0.49170	0.05568	0.04802	0.14606	0.24206
127	0.49205	0.05611	0.04795	0.14602	0.24202	183	0.49094	0.05579	0.04779	0.14588	0.24187
128	0.49123	0.05581	0.04824	0.14601	0.24202	184	0.49203	0.05573	0.04798	0.14637	0.24230
129	0.49125	0.05588	0.04810	0.14603	0.24202	185	0.49203	0.05616	0.04800	0.14590	0.24192
130	0.49082	0.05619	0.04766	0.14620	0.24216	186	0.49084	0.05581	0.04814	0.14578	0.24182
131	0.49230	0.05611	0.04767	0.14583	0.24187	187	0.49135	0.05593	0.04809	0.14602	0.24202
132	0.49152	0.05544	0.04843	0.14600	0.24202	188	0.49128	0.05642	0.04803	0.14562	0.24168
133	0.49199	0.05607	0.04828	0.14572	0.24178	189	0.49165	0.05622	0.04772	0.14601	0.24202
134	0.49174	0.05646	0.04748	0.14607	0.24206	190	0.49129	0.05630	0.04804	0.14573	0.24178
135	0.49171	0.05614	0.04763	0.14605	0.24206	191	0.49204	0.05602	0.04819	0.14588	0.24192
136	0.49204	0.05603	0.04784	0.14620	0.24216	192	0.49303	0.05601	0.04803	0.14594	0.24197
137	0.49165	0.05623	0.04773	0.14604	0.24206	193	0.49148	0.05605	0.04775	0.14617	0.24216
138	0.49019	0.05598	0.04811	0.14595	0.24197	194	0.49115	0.05626	0.04771	0.14608	0.24206
139	0.49125	0.05612	0.04769	0.14595	0.24197	195	0.49205	0.05616	0.04807	0.14585	0.24187
140	0.49096	0.05618	0.04778	0.14610	0.24211	196	0.49097	0.05607	0.04770	0.14623	0.24221
141	0.49127	0.05578	0.04825	0.14586	0.24192	197	0.49197	0.05612	0.04755	0.14621	0.24221
142	0.49130	0.05568	0.04768	0.14671	0.24259	198	0.49121	0.05567	0.04774	0.14614	0.24211
143	0.49146	0.05586	0.04827	0.14593	0.24197	199	0.49096	0.05615	0.04768	0.14612	0.24211
144	0.49102	0.05618	0.04787	0.14599	0.24202	200	0.49135	0.05592	0.04785	0.14622	0.24221
145	0.49170	0.05602	0.04775	0.14628	0.24226	201	0.49233	0.05601	0.04804	0.14604	0.24206
146	0.49206	0.05608	0.04806	0.14594	0.24197	202	0.49101	0.05641	0.04785	0.14580	0.24187
147	0.49128	0.05632	0.04793	0.14565	0.24173	203	0.49141	0.05600	0.04781	0.14623	0.24221
148	0.49254	0.05650	0.04776	0.14577	0.24182	204	0.49200	0.05625	0.04788	0.14595	0.24197
149	0.49031	0.05656	0.04732	0.14591	0.24192	205	0.49132	0.05604	0.04788	0.14607	0.24206
150	0.49110	0.05607	0.04804	0.14593	0.24197	206	0.49120	0.05610	0.04791	0.14605	0.24206
151	0.49130	0.05606	0.04811	0.14587	0.24192	207	0.49198	0.05617	0.04788	0.14599	0.24202
152	0.49150	0.05598	0.04769	0.14640	0.24235	208	0.49200	0.05620	0.04783	0.14605	0.24206
153	0.49185	0.05613	0.04817	0.14573	0.24178	209	0.49155	0.05614	0.04796	0.14596	0.24197
154	0.49223	0.05575	0.04820	0.14603	0.24206	210	0.49211	0.05647	0.04755	0.14584	0.24187
155	0.49176	0.05605	0.04800	0.14603	0.24206	211	0.49166	0.05616	0.04605	0.14554	0.24187
156	0.49181	0.05674	0.04812	0.14603	0.24206	212	0.49170	0.05627	0.04812	0.14561	0.24168
157	0.49139	0.05594	0.04809	0.14601	0.24202	213	0.49133	0.05629	0.04771	0.14587	0.24192
158	0.49202	0.05663	0.04767	0.14575	0.24182	214	0.49208	0.05568	0.04796	0.14622	0.24221
159	0.49160	0.05614	0.04779	0.14605	0.24206	215	0.49184	0.05600	0.04836	0.14572	0.24178
160	0.49090	0.05623	0.04755	0.14575	0.24182	216	0.49077	0.05620	0.04783	0.14600	0.24202
161	0.49135	0.05595	0.04627	0.14585	0.24187	217	0.49160	0.05574	0.04793	0.14638	0.24235
162	0.49218	0.05632	0.04785	0.14584	0.24187	218	0.49152	0.05623	0.04811	0.14573	0.24178
163	0.49187	0.05561	0.04833	0.14804	0.24206	219	0.49246	0.05588	0.04765	0.14597	0.24197
164	0.49144	0.05617	0.04789	0.14602	0.24202	220	0.49186	0.05615	0.04789	0.14596	0.24197
165	0.49275	0.05621	0.04809	0.14577	0.24182	221	0.49146	0.05628	0.04794	0.14586	0.24192
166	0.49032	0.05624	0.04790	0.14579	0.24182	222	0.49178	0.05568	0.04807	0.14633	0.24230
167	0.49183	0.05597	0.04803	0.14608	0.24206	223	0.49156	0.05561	0.04850	0.14568	0.24176
168	0.49181	0.05624	0.04781	0.14601		224	0.49173	0.05610	0.04798	0.14599	0.24202

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่ผู้แบบเบตริกซ์จังหวัดคุณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน											
No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
225	0.49219	0.05561	0.04797	0.14622	0.24221	281	0.49182	0.05619	0.04601	0.14585	0.24187
226	0.49126	0.05619	0.04818	0.14668	0.24173	282	0.49210	0.05619	0.04793	0.14590	0.24192
227	0.49180	0.05643	0.04745	0.14578	0.24182	283	0.49186	0.05585	0.04799	0.14621	0.24221
228	0.49117	0.05585	0.04811	0.14609	0.24211	284	0.49119	0.05591	0.04807	0.14608	0.24206
229	0.49231	0.05618	0.04783	0.14603	0.24202	285	0.49168	0.05617	0.04804	0.14587	0.24192
230	0.49309	0.05620	0.04770	0.14601	0.24202	286	0.49197	0.05609	0.04774	0.14613	0.24211
231	0.49245	0.05586	0.04791	0.14621	0.24221	287	0.49181	0.05598	0.04792	0.14618	0.24216
232	0.49056	0.05620	0.04740	0.14589	0.24192	288	0.49230	0.05611	0.04779	0.14594	0.24197
233	0.49183	0.05620	0.04769	0.14575	0.24182	289	0.49199	0.05634	0.04790	0.14584	0.24187
234	0.49120	0.05583	0.04798	0.14623	0.24221	290	0.49132	0.05587	0.04817	0.14603	0.24202
235	0.49195	0.05609	0.04795	0.14605	0.24206	291	0.49206	0.05595	0.04728	0.14546	0.24158
236	0.49153	0.05555	0.04773	0.14588	0.24173	292	0.49187	0.05607	0.04805	0.14592	0.24197
237	0.49176	0.05639	0.04788	0.14581	0.24187	293	0.49155	0.05613	0.04761	0.14607	0.24206
238	0.49150	0.05596	0.04798	0.14610	0.24211	294	0.49205	0.05630	0.04718	0.14611	0.24211
239	0.49238	0.05574	0.04803	0.14608	0.24206	295	0.49246	0.05608	0.04834	0.14560	0.24168
240	0.49184	0.05637	0.04784	0.14585	0.24187	296	0.49211	0.05630	0.04801	0.14571	0.24178
241	0.49078	0.05588	0.04806	0.14599	0.24202	297	0.49229	0.05587	0.04824	0.14597	0.24197
242	0.49239	0.05635	0.04800	0.14567	0.24173	298	0.49145	0.05577	0.04815	0.14604	0.24206
243	0.49102	0.05617	0.04809	0.14579	0.24182	299	0.49147	0.05643	0.04769	0.14562	0.24168
244	0.49077	0.05616	0.04800	0.14589	0.24192	300	0.49143	0.05591	0.04820	0.14594	0.24197
245	0.49120	0.05596	0.04799	0.14612	0.24211	301	0.49205	0.05614	0.04790	0.14604	0.24206
246	0.49070	0.05630	0.04770	0.14606	0.24206	302	0.49123	0.05601	0.04806	0.14600	0.24202
247	0.49233	0.05623	0.04764	0.14602	0.24202	303	0.49068	0.05609	0.04766	0.14604	0.24206
248	0.49120	0.05629	0.04793	0.14584	0.24187	304	0.49180	0.05587	0.04818	0.14603	0.24202
249	0.49152	0.05579	0.04610	0.14604	0.24206	305	0.49123	0.05640	0.04707	0.14569	0.24192
250	0.49267	0.05571	0.04837	0.14601	0.24202	306	0.49219	0.05590	0.04814	0.14580	0.24187
251	0.49142	0.05591	0.04805	0.14587	0.24192	307	0.49151	0.05608	0.04788	0.14610	0.24211
252	0.49136	0.05630	0.04800	0.14576	0.24182	308	0.49197	0.05618	0.04779	0.14608	0.24206
253	0.49163	0.05602	0.04782	0.14624	0.24221	309	0.49176	0.05581	0.04787	0.14626	0.24221
254	0.49065	0.05585	0.04809	0.14590	0.24192	310	0.49214	0.05580	0.04806	0.14610	0.24210
255	0.49220	0.05564	0.04828	0.14606	0.24206	311	0.49120	0.05574	0.04625	0.14606	0.24206
256	0.49162	0.05610	0.04779	0.14592	0.24197	312	0.49081	0.05640	0.04765	0.14601	0.24202
257	0.49217	0.05618	0.04787	0.14576	0.24182	313	0.49154	0.05614	0.04809	0.14584	0.24187
258	0.49232	0.05628	0.04797	0.14555	0.24163	314	0.49178	0.05577	0.04816	0.14614	0.24211
259	0.49175	0.05612	0.04784	0.14597	0.24202	315	0.49236	0.05622	0.04762	0.14604	0.24206
260	0.49172	0.05598	0.04803	0.14607	0.24206	316	0.49162	0.05585	0.04801	0.14620	0.24216
261	0.49104	0.05619	0.04600	0.14583	0.24187	317	0.49146	0.05642	0.04702	0.14576	0.24182
262	0.49177	0.05607	0.04800	0.14600	0.24202	318	0.49175	0.05595	0.04753	0.14573	0.24178
263	0.49204	0.05592	0.04846	0.14570	0.24178	319	0.49161	0.05609	0.04789	0.14610	0.24211
264	0.49144	0.05580	0.04798	0.14626	0.24221	320	0.49166	0.05607	0.04765	0.14632	0.24226
265	0.49167	0.05582	0.04809	0.14605	0.24206	321	0.49149	0.05616	0.04766	0.14624	0.24221
266	0.49115	0.05599	0.04804	0.14595	0.24197	322	0.49198	0.05618	0.04803	0.14571	0.24178
267	0.49201	0.05632	0.04765	0.14591	0.24192	323	0.49070	0.05612	0.04718	0.14575	0.24182
268	0.49137	0.05584	0.04788	0.14658	0.24250	324	0.49136	0.05585	0.04804	0.14616	0.24216
269	0.49134	0.05590	0.04793	0.14621	0.24221	325	0.49118	0.05592	0.04781	0.14633	0.24230
270	0.49206	0.05605	0.04802	0.14601	0.24202	326	0.49235	0.05626	0.04796	0.14584	0.24187
271	0.49145	0.05623	0.04805	0.14579	0.24182	327	0.49117	0.05627	0.04764	0.14615	0.24211
272	0.49161	0.05602	0.04777	0.14622	0.24221	328	0.49153	0.05637	0.04780	0.14585	0.24187
273	0.49242	0.05587	0.04806	0.14606	0.24206	329	0.49120	0.05651	0.04774	0.14562	0.24167
274	0.49169	0.05594	0.04758	0.14645	0.24240	330	0.49081	0.05632	0.04782	0.14586	0.24187
275	0.49124	0.05572	0.04799	0.14621	0.24221	331	0.49134	0.05613	0.04785	0.14597	0.24197
276	0.49113	0.05599	0.04806	0.14600	0.24202	332	0.49256	0.05602	0.04814	0.14589	0.24192
277	0.49093	0.05601	0.04776	0.14618	0.24216	333	0.49131	0.05573	0.04819	0.14614	0.24211
278	0.49165	0.05611	0.04778	0.14609	0.24211	334	0.49146	0.05583	0.04797	0.14597	0.24202
279	0.49142	0.05589	0.04795	0.14623	0.24221	335	0.49260	0.05578	0.04803	0.14612	0.24211
280	0.49127	0.05643	0.04760	0.14576	0.24182	336	0.49150	0.05568	0.04823	0.14615	0.24216

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่ผู้มาประเมินตัวชี้วัดพึงพอใจ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 500 คน											
No.	Mean	VarE	VarI	VarE!	S.E.Mean	No.	Mean	VarE	VarI	VarE!	S.E.Mean
337	0.49164	0.05616	0.04771	0.14602	0.24202	395	0.49190	0.05583	0.04795	0.14607	0.24206
338	0.49203	0.05614	0.04737	0.14610	0.24211	394	0.49170	0.05586	0.04808	0.14609	0.24211
339	0.49135	0.05638	0.04771	0.14590	0.24192	395	0.49157	0.05570	0.04811	0.14588	0.24192
340	0.49166	0.05583	0.04819	0.14604	0.24206	396	0.49195	0.05606	0.04765	0.14618	0.24216
341	0.49219	0.05589	0.04812	0.14564	0.24173	397	0.49223	0.05588	0.04822	0.14586	0.24192
342	0.49184	0.05650	0.04760	0.14592	0.24197	398	0.49206	0.05609	0.04824	0.14569	0.24178
343	0.49079	0.05625	0.04770	0.14564	0.24173	399	0.49095	0.05602	0.04770	0.14616	0.24216
344	0.49055	0.05584	0.04763	0.14617	0.24216	400	0.49190	0.05613	0.04804	0.14579	0.24182
345	0.49270	0.05576	0.04804	0.14623	0.24221	401	0.49159	0.05622	0.04785	0.14580	0.24187
346	0.49182	0.05608	0.04778	0.14621	0.24221	402	0.49207	0.05610	0.04812	0.14585	0.24187
347	0.49212	0.05569	0.04805	0.14581	0.24187	403	0.49111	0.05628	0.04757	0.14613	0.24211
348	0.49143	0.05627	0.04810	0.14589	0.24176	404	0.49117	0.05616	0.04783	0.14590	0.24192
349	0.49172	0.05609	0.04831	0.14563	0.24173	405	0.49085	0.05570	0.04799	0.14564	0.24187
350	0.49178	0.05606	0.04799	0.14596	0.24197	406	0.49065	0.05616	0.04753	0.14591	0.24192
351	0.49147	0.05594	0.04820	0.14568	0.24173	407	0.49188	0.05636	0.04799	0.14573	0.24178
352	0.49177	0.05583	0.04848	0.14573	0.24178	408	0.49180	0.05605	0.04809	0.14595	0.24197
353	0.49188	0.05610	0.04803	0.14594	0.24197	409	0.49142	0.05581	0.04804	0.14612	0.24211
354	0.49099	0.05633	0.04800	0.14573	0.24172	410	0.49241	0.05530	0.04775	0.14591	0.24192
355	0.49177	0.05642	0.04781	0.14530	0.24144	411	0.49080	0.05563	0.04617	0.14606	0.24206
356	0.49214	0.05607	0.04811	0.14591	0.24192	412	0.49115	0.05584	0.04788	0.14610	0.24211
357	0.49146	0.05605	0.04786	0.14615	0.24216	413	0.49222	0.05586	0.04790	0.14614	0.24211
358	0.49120	0.05585	0.04804	0.14558	0.24168	414	0.49204	0.05639	0.04766	0.14602	0.24202
359	0.49198	0.05608	0.04786	0.14592	0.24197	415	0.49161	0.05631	0.04787	0.14587	0.24192
360	0.49123	0.05603	0.04763	0.14612	0.24211	416	0.49109	0.05618	0.04767	0.14622	0.24221
361	0.49219	0.05615	0.04784	0.14609	0.24211	417	0.49184	0.05593	0.04755	0.14609	0.24206
362	0.49167	0.05596	0.04794	0.14603	0.24206	418	0.49200	0.05611	0.04816	0.14580	0.24187
363	0.49190	0.05615	0.04791	0.14602	0.24202	419	0.49104	0.05618	0.04833	0.14554	0.24163
364	0.49158	0.05597	0.04794	0.14613	0.24211	420	0.49167	0.05631	0.04794	0.14578	0.24182
365	0.49156	0.05626	0.04724	0.14588	0.24192	421	0.49157	0.05607	0.04828	0.14563	0.24173
366	0.49143	0.05593	0.04810	0.14587	0.24192	422	0.49120	0.05621	0.04790	0.14596	0.24197
367	0.49151	0.05577	0.04799	0.14632	0.24226	423	0.49119	0.05646	0.04764	0.14570	0.24178
368	0.49125	0.05600	0.04796	0.14605	0.24206	424	0.49228	0.05616	0.04790	0.14595	0.24197
369	0.49151	0.05606	0.04792	0.14609	0.24211	425	0.49164	0.05608	0.04796	0.14602	0.24202
370	0.49187	0.05634	0.04738	0.14604	0.24206	426	0.49166	0.05614	0.04785	0.14607	0.24206
371	0.49178	0.05617	0.04796	0.14586	0.24187	427	0.49169	0.05602	0.04799	0.14578	0.24182
372	0.49177	0.05635	0.04784	0.14586	0.24192	428	0.49258	0.05630	0.04793	0.14580	0.24178
373	0.49168	0.05567	0.04791	0.14616	0.24216	429	0.49151	0.05600	0.04767	0.14610	0.24211
374	0.49209	0.05585	0.04807	0.14610	0.24211	430	0.49121	0.05598	0.04817	0.14576	0.24182
375	0.49115	0.05619	0.04769	0.14608	0.24206	431	0.49106	0.05614	0.04796	0.14588	0.24192
376	0.49210	0.05575	0.04818	0.14603	0.24202	432	0.49153	0.05609	0.04799	0.14598	0.24202
377	0.49235	0.05642	0.04761	0.14599	0.24202	433	0.49244	0.05622	0.04806	0.14589	0.24178
378	0.49100	0.05612	0.04797	0.14592	0.24197	434	0.49141	0.05627	0.04782	0.14584	0.24187
379	0.49142	0.05601	0.04801	0.14593	0.24197	435	0.49170	0.05616	0.04793	0.14596	0.24197
380	0.49124	0.05611	0.04805	0.14591	0.24192	436	0.49173	0.05584	0.04808	0.14610	0.24211
381	0.49180	0.05608	0.04822	0.14552	0.24163	437	0.49114	0.05560	0.04817	0.14618	0.24216
382	0.49198	0.05613	0.04770	0.14594	0.24197	438	0.49044	0.05600	0.04794	0.14568	0.24173
383	0.49085	0.05585	0.04830	0.14591	0.24192	439	0.49210	0.05600	0.04771	0.14621	0.24221
384	0.49115	0.05602	0.04780	0.14612	0.24211	440	0.49216	0.05598	0.04821	0.14567	0.24173
385	0.49026	0.05610	0.04795	0.14573	0.24178	441	0.49214	0.05614	0.04801	0.14575	0.24182
386	0.49054	0.05583	0.04787	0.14559	0.24168	442	0.49118	0.05638	0.04781	0.14587	0.24192
387	0.49164	0.05613	0.04798	0.14587	0.24192	443	0.49189	0.05558	0.04788	0.14639	0.24235
388	0.49175	0.05594	0.04841	0.14573	0.24178	444	0.49207	0.05626	0.04774	0.14596	0.24197
389	0.49063	0.05593	0.04821	0.14591	0.24192	445	0.49125	0.05609	0.04787	0.14611	0.24211
390	0.49205	0.05588	0.04822	0.14588	0.24192	446	0.49177	0.05603	0.04773	0.14613	0.24211
391	0.49101	0.05609	0.04804	0.14586	0.24192	447	0.49178	0.05611	0.04789	0.14596	0.24202
392	0.49164	0.05623	0.04775	0.14605	0.24206	448	0.49151	0.05574	0.04774	0.14641	0.24235

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่มแร ๆ เมตริกที่เชิงพหุคูณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน											
No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
449	0.49151	0.05623	0.04605	0.14576	0.24182	505	0.49212	0.05576	0.04780	0.14625	0.24221
450	0.49206	0.05681	0.04790	0.14632	0.24230	506	0.49245	0.05621	0.04823	0.14560	0.24168
451	0.49173	0.05615	0.04789	0.14602	0.24202	507	0.49196	0.05654	0.04762	0.14589	0.24192
452	0.49129	0.05611	0.04786	0.14603	0.24202	508	0.49184	0.05585	0.04790	0.14633	0.24230
453	0.49239	0.05663	0.04802	0.14550	0.24158	509	0.49201	0.05629	0.04793	0.14586	0.24192
454	0.49085	0.05600	0.04801	0.14501	0.24192	510	0.49273	0.05652	0.04772	0.14585	0.24187
455	0.49109	0.05610	0.04786	0.14600	0.24202	511	0.49175	0.05584	0.04766	0.14622	0.24221
456	0.49155	0.05609	0.04789	0.14606	0.24206	512	0.49137	0.05622	0.04748	0.14637	0.24230
457	0.49121	0.05584	0.04801	0.14615	0.24211	513	0.49195	0.05594	0.04813	0.14599	0.24202
458	0.49294	0.05581	0.04809	0.14620	0.24216	514	0.49222	0.05598	0.04788	0.14622	0.24221
459	0.49055	0.05589	0.04808	0.14583	0.24187	515	0.49162	0.05611	0.04769	0.14627	0.24226
460	0.49123	0.05620	0.04763	0.14577	0.24182	516	0.49147	0.05597	0.04803	0.14606	0.24206
461	0.49197	0.05584	0.04812	0.14594	0.24197	517	0.49130	0.05620	0.04804	0.14580	0.24187
462	0.49111	0.05581	0.04798	0.14617	0.24216	518	0.49160	0.05578	0.04788	0.14618	0.24216
463	0.49161	0.05614	0.04814	0.14578	0.24182	519	0.49112	0.05594	0.04787	0.14620	0.24216
464	0.49122	0.05623	0.04805	0.14579	0.24182	520	0.49187	0.05602	0.04784	0.14623	0.24221
465	0.49186	0.05597	0.04784	0.14635	0.24230	521	0.49116	0.05594	0.04770	0.14639	0.24230
466	0.49102	0.05615	0.04756	0.14601	0.24202	522	0.49138	0.05650	0.04768	0.14573	0.24170
467	0.49169	0.05623	0.04771	0.14606	0.24206	523	0.49190	0.05567	0.04820	0.14621	0.24221
468	0.49150	0.05638	0.04789	0.14578	0.24182	524	0.49243	0.05620	0.04765	0.14599	0.24202
469	0.49195	0.05589	0.04842	0.14558	0.24163	525	0.49137	0.05591	0.04818	0.14561	0.24168
470	0.49135	0.05590	0.04790	0.14627	0.24226	526	0.49152	0.05630	0.04795	0.14579	0.24182
471	0.49169	0.05607	0.04789	0.14609	0.24206	527	0.49199	0.05592	0.04809	0.14605	0.24206
472	0.49041	0.05632	0.04761	0.14579	0.24182	528	0.49180	0.05601	0.04811	0.14593	0.24187
473	0.49068	0.05606	0.04762	0.14605	0.24206	529	0.49151	0.05625	0.04773	0.14609	0.24211
474	0.49165	0.05593	0.04817	0.14597	0.24197	530	0.49176	0.05576	0.04794	0.14638	0.24236
475	0.49230	0.05592	0.04794	0.14610	0.24211	531	0.49102	0.05622	0.04754	0.14626	0.24221
476	0.49159	0.05613	0.04777	0.14606	0.24206	532	0.49131	0.05572	0.04771	0.14636	0.24230
477	0.49170	0.05612	0.04794	0.14597	0.24197	533	0.49138	0.05634	0.04782	0.14592	0.24197
478	0.49141	0.05617	0.04790	0.14607	0.24206	534	0.49197	0.05599	0.04785	0.14602	0.24202
479	0.49241	0.05634	0.04807	0.14602	0.24202	535	0.49112	0.05596	0.04779	0.14632	0.24226
480	0.49262	0.05616	0.04815	0.14570	0.24178	536	0.49110	0.05607	0.04801	0.14598	0.24202
481	0.49207	0.05625	0.04787	0.14593	0.24197	537	0.49236	0.05606	0.04791	0.14612	0.24211
482	0.49171	0.05606	0.04752	0.14584	0.24187	538	0.49089	0.05661	0.04746	0.14567	0.24173
483	0.49221	0.05595	0.04801	0.14595	0.24197	539	0.49073	0.05594	0.04817	0.14594	0.24197
484	0.49085	0.05622	0.04750	0.14575	0.24182	540	0.49147	0.05574	0.04822	0.14612	0.24211
485	0.49141	0.05610	0.04774	0.14618	0.24216	541	0.49180	0.05613	0.04765	0.14628	0.24226
486	0.49186	0.05573	0.04822	0.14611	0.24211	542	0.49107	0.05597	0.04783	0.14591	0.24192
487	0.49122	0.05620	0.04783	0.14601	0.24202	543	0.49145	0.05604	0.04786	0.14616	0.24216
488	0.49185	0.05622	0.04781	0.14563	0.24173	544	0.49164	0.05606	0.04835	0.14555	0.24163
489	0.49101	0.05588	0.04790	0.14628	0.24226	545	0.49102	0.05610	0.04796	0.14600	0.24202
490	0.49059	0.05571	0.04816	0.14602	0.24202	546	0.49121	0.05628	0.04802	0.14577	0.24182
491	0.49166	0.05598	0.04766	0.14616	0.24216	547	0.49152	0.05604	0.04817	0.14584	0.24187
492	0.49081	0.05612	0.04738	0.14652	0.24245	548	0.49137	0.05621	0.04759	0.14617	0.24216
493	0.49253	0.05557	0.04813	0.14597	0.24197	549	0.49214	0.05578	0.04810	0.14598	0.24202
494	0.49136	0.05622	0.04786	0.14591	0.24192	550	0.49232	0.05553	0.04843	0.14611	0.24211
495	0.49151	0.05587	0.04810	0.14610	0.24211	551	0.49212	0.05671	0.04760	0.14577	0.24182
496	0.49190	0.05610	0.04774	0.14573	0.24178	552	0.49172	0.05638	0.04759	0.14558	0.24168
497	0.49231	0.05594	0.04814	0.14601	0.24202	553	0.49135	0.05601	0.04778	0.14615	0.24216
498	0.49242	0.05612	0.04803	0.14593	0.24197	554	0.49127	0.05617	0.04752	0.14634	0.24230
499	0.49255	0.05572	0.04793	0.14632	0.24230	555	0.49129	0.05583	0.04769	0.14846	0.24240
500	0.49197	0.05599	0.04769	0.14578	0.24182	556	0.49152	0.05589	0.04801	0.14587	0.24192
501	0.49213	0.05606	0.04796	0.14605	0.24206	557	0.49089	0.05590	0.04772	0.14609	0.24211
502	0.49187	0.05611	0.04780	0.14636	0.24230	558	0.49104	0.05661	0.04767	0.14559	0.24168
503	0.49217	0.05579	0.04825	0.14600	0.24202	559	0.49153	0.05571	0.04796	0.14621	0.24221
504	0.49128	0.05615	0.04807	0.14585	0.24187	560	0.49144	0.05634	0.04772	0.14595	0.24197

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่มแบบเมตริกซ์เชิงพหุคูณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน											
No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
561	0.49137	0.05593	0.04798	0.14611	0.24211	617	0.49136	0.05598	0.04805	0.14599	0.24202
562	0.49088	0.05632	0.04820	0.14555	0.24163	618	0.49091	0.05573	0.04796	0.14622	0.24221
563	0.49215	0.05584	0.04804	0.14614	0.24211	619	0.49182	0.05582	0.04803	0.14617	0.24216
564	0.49135	0.05613	0.04775	0.14613	0.24211	620	0.49181	0.05627	0.04797	0.14584	0.24187
565	0.49154	0.05621	0.04748	0.14538	0.24149	621	0.49163	0.05587	0.04815	0.14584	0.24187
566	0.49193	0.05622	0.04792	0.14592	0.24197	622	0.49195	0.05588	0.04779	0.14591	0.24192
567	0.49085	0.05619	0.04790	0.14594	0.24197	623	0.49139	0.05625	0.04809	0.14572	0.24178
568	0.49064	0.05632	0.04805	0.14569	0.24178	624	0.49226	0.05599	0.04814	0.14591	0.24192
569	0.49234	0.05608	0.04780	0.14618	0.24216	625	0.49227	0.05616	0.04797	0.14585	0.24187
570	0.49161	0.05563	0.04787	0.14601	0.24202	626	0.49109	0.05597	0.04819	0.14586	0.24192
571	0.49156	0.05616	0.04805	0.14576	0.24182	627	0.49260	0.05579	0.04814	0.14607	0.24206
572	0.49045	0.05619	0.04822	0.14560	0.24168	628	0.49143	0.05645	0.04781	0.14581	0.24187
573	0.49168	0.05598	0.04818	0.14588	0.24192	629	0.49097	0.05650	0.04799	0.14556	0.24163
574	0.49135	0.05618	0.04776	0.14601	0.24202	630	0.49181	0.05584	0.04753	0.14625	0.24221
575	0.49162	0.05600	0.04791	0.14611	0.24211	631	0.49121	0.05591	0.04787	0.14599	0.24202
576	0.49122	0.05645	0.04776	0.14562	0.24168	632	0.49190	0.05571	0.04795	0.14638	0.24230
577	0.49206	0.05598	0.04776	0.14599	0.24202	633	0.49190	0.05606	0.04784	0.14611	0.24211
578	0.49146	0.05603	0.04784	0.14599	0.24202	634	0.49188	0.05582	0.04796	0.14625	0.24221
579	0.49092	0.05595	0.04801	0.14611	0.24211	635	0.49111	0.05583	0.04830	0.14592	0.24197
580	0.49176	0.05540	0.04797	0.14656	0.24250	636	0.49201	0.05621	0.04802	0.14584	0.24187
581	0.49214	0.05596	0.04779	0.14627	0.24226	637	0.49125	0.05573	0.04778	0.14651	0.24245
582	0.49180	0.05613	0.04785	0.14605	0.24206	638	0.49196	0.05643	0.04736	0.14615	0.24211
583	0.49252	0.05598	0.04771	0.14628	0.24226	639	0.49236	0.05594	0.04778	0.14602	0.24202
584	0.49101	0.05614	0.04819	0.14568	0.24173	640	0.49201	0.05628	0.04812	0.14566	0.24173
585	0.49125	0.05604	0.04821	0.14574	0.24178	641	0.49089	0.05640	0.04746	0.14604	0.24206
586	0.49175	0.05633	0.04789	0.14585	0.24187	642	0.49195	0.05623	0.04814	0.14571	0.24178
587	0.49121	0.05629	0.04780	0.14594	0.24197	643	0.49265	0.05601	0.04767	0.14633	0.24230
588	0.49184	0.05632	0.04784	0.14583	0.24187	644	0.49186	0.05633	0.04775	0.14559	0.24168
589	0.49049	0.05585	0.04796	0.14624	0.24221	645	0.49235	0.05615	0.04817	0.14577	0.24182
590	0.49115	0.05618	0.04773	0.14586	0.24192	646	0.49231	0.05651	0.04769	0.14584	0.24187
591	0.49044	0.05614	0.04731	0.14579	0.24182	647	0.49215	0.05623	0.04715	0.14656	0.24250
592	0.49103	0.05612	0.04772	0.14571	0.24178	648	0.49081	0.05616	0.04770	0.14617	0.24216
593	0.49131	0.05572	0.04827	0.14608	0.24206	649	0.49161	0.05622	0.04763	0.14598	0.24202
594	0.49159	0.05601	0.04795	0.14610	0.24211	650	0.49272	0.05613	0.04830	0.14539	0.24149
595	0.49211	0.05595	0.04812	0.14593	0.24197	651	0.49137	0.05618	0.04800	0.14574	0.24182
596	0.49203	0.05615	0.04785	0.14607	0.24206	652	0.49185	0.05602	0.04807	0.14595	0.24197
597	0.49128	0.05579	0.04808	0.14618	0.24216	653	0.49125	0.05640	0.04754	0.14592	0.24197
598	0.49115	0.05632	0.04773	0.14601	0.24202	654	0.49134	0.05582	0.04809	0.14613	0.24211
599	0.49182	0.05607	0.04759	0.14608	0.24206	655	0.49153	0.05609	0.04802	0.14590	0.24192
600	0.49174	0.05554	0.04810	0.14632	0.24226	656	0.49238	0.05573	0.04809	0.14602	0.24202
601	0.49092	0.05616	0.04785	0.14591	0.24192	657	0.49094	0.05602	0.04801	0.14599	0.24202
602	0.49159	0.05597	0.04835	0.14574	0.24182	658	0.49266	0.05593	0.04771	0.14635	0.24230
603	0.49122	0.05562	0.04835	0.14610	0.24211	659	0.49168	0.05605	0.04789	0.14612	0.24211
604	0.49154	0.05586	0.04788	0.14631	0.24226	660	0.49133	0.05591	0.04791	0.14593	0.24197
605	0.49180	0.05656	0.04748	0.14570	0.24178	661	0.49179	0.05575	0.04816	0.14604	0.24206
606	0.49140	0.05645	0.04786	0.14575	0.24182	662	0.49167	0.05604	0.04737	0.14588	0.24192
607	0.49195	0.05641	0.04777	0.14590	0.24192	663	0.49169	0.05591	0.04809	0.14565	0.24173
608	0.49241	0.05603	0.04829	0.14567	0.24173	664	0.49154	0.05603	0.04809	0.14594	0.24197
609	0.49193	0.05627	0.04792	0.14589	0.24192	665	0.49199	0.05633	0.04773	0.14577	0.24182
610	0.49193	0.05613	0.04749	0.14642	0.24235	666	0.49065	0.05697	0.04785	0.14597	0.24197
611	0.49125	0.05590	0.04782	0.14621	0.24221	667	0.49155	0.05611	0.04777	0.14608	0.24206
612	0.49061	0.05592	0.04771	0.14596	0.24197	668	0.49189	0.05632	0.04706	0.14621	0.24221
613	0.49186	0.05589	0.04811	0.14606	0.24206	669	0.49219	0.05597	0.04830	0.14573	0.24178
614	0.49104	0.05598	0.04784	0.14619	0.24216	670	0.49246	0.05640	0.04770	0.14598	0.24202
615	0.49168	0.05594	0.04798	0.14607	0.24206	671	0.49248	0.05607	0.04806	0.14597	0.24197
616	0.49147	0.05610	0.04800	0.14593	0.24197	672	0.49218	0.05636	0.04744	0.14617	0.24216

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่มแบบเมตริกซ์เชิงพหุคูณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน

No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
673	0.49167	0.05591	0.04784	0.14614	0.24211	729	0.49141	0.05594	0.04769	0.14636	0.24230
674	0.49119	0.05612	0.04794	0.14596	0.24197	730	0.49095	0.05611	0.04828	0.14567	0.24173
675	0.49173	0.05637	0.04792	0.14579	0.24182	731	0.49158	0.05605	0.04819	0.14583	0.24187
676	0.49191	0.05621	0.04819	0.14568	0.24173	732	0.49169	0.05595	0.04801	0.14609	0.24211
677	0.49183	0.05600	0.04809	0.14569	0.24178	733	0.49090	0.05642	0.04768	0.14591	0.24192
678	0.49084	0.05609	0.04804	0.14592	0.24197	734	0.49163	0.05626	0.04745	0.14569	0.24178
679	0.49182	0.05605	0.04797	0.14579	0.24182	735	0.49128	0.05597	0.04799	0.14600	0.24202
680	0.49127	0.05576	0.04820	0.14611	0.24211	736	0.49266	0.05603	0.04771	0.14627	0.24226
681	0.49204	0.05611	0.04791	0.14584	0.24187	737	0.49202	0.05587	0.04766	0.14644	0.24240
682	0.49056	0.05634	0.04755	0.14589	0.24192	738	0.49216	0.05602	0.04786	0.14619	0.24216
683	0.49232	0.05590	0.04810	0.14594	0.24197	739	0.49161	0.05611	0.04741	0.14653	0.24245
684	0.49224	0.05618	0.04773	0.14605	0.24206	740	0.49158	0.05592	0.04820	0.14596	0.24197
685	0.49136	0.05595	0.04810	0.14602	0.24202	741	0.49082	0.05655	0.04765	0.14580	0.24187
686	0.49139	0.05595	0.04771	0.14620	0.24216	742	0.49198	0.05614	0.04794	0.14591	0.24192
687	0.49119	0.05647	0.04757	0.14587	0.24192	743	0.49194	0.05597	0.04818	0.14589	0.24192
688	0.49191	0.05600	0.04749	0.14646	0.24240	744	0.49184	0.05596	0.04791	0.14620	0.24216
689	0.49059	0.05579	0.04825	0.14601	0.24202	745	0.49201	0.05586	0.04838	0.14582	0.24187
690	0.49191	0.05627	0.04760	0.14604	0.24206	746	0.49163	0.05591	0.04835	0.14563	0.24168
691	0.49131	0.05570	0.04824	0.14599	0.24202	747	0.49169	0.05555	0.04800	0.14619	0.24216
692	0.49138	0.05632	0.04783	0.14588	0.24192	748	0.49227	0.05596	0.04794	0.14618	0.24216
693	0.49076	0.05624	0.04791	0.14591	0.24192	749	0.49190	0.05615	0.04800	0.14593	0.24197
694	0.49078	0.05627	0.04782	0.14595	0.24197	750	0.49252	0.05607	0.04795	0.14599	0.24202
695	0.49216	0.05638	0.04775	0.14586	0.24187	751	0.49207	0.05619	0.04785	0.14595	0.24197
696	0.49160	0.05612	0.04807	0.14588	0.24192	752	0.49186	0.05570	0.04806	0.14605	0.24206
697	0.49170	0.05652	0.04749	0.14604	0.24206	753	0.49139	0.05592	0.04807	0.14602	0.24202
698	0.49142	0.05623	0.04832	0.14553	0.24163	754	0.49231	0.05616	0.04806	0.14582	0.24187
699	0.49129	0.05571	0.04825	0.14588	0.24202	755	0.49282	0.05611	0.04786	0.14610	0.24211
700	0.49107	0.05589	0.04809	0.14608	0.24206	756	0.49072	0.05606	0.04785	0.14615	0.24211
701	0.49092	0.05587	0.04796	0.14600	0.24202	757	0.49123	0.05601	0.04762	0.14599	0.24202
702	0.49239	0.05582	0.04844	0.14582	0.24187	758	0.49144	0.05636	0.04757	0.14576	0.24182
703	0.49195	0.05603	0.04752	0.14652	0.24245	759	0.49203	0.05597	0.04769	0.14634	0.24230
704	0.49188	0.05597	0.04844	0.14562	0.24168	760	0.49207	0.05585	0.04773	0.14650	0.24245
705	0.49157	0.05571	0.04817	0.14587	0.24192	761	0.49186	0.05605	0.04805	0.14592	0.24197
706	0.49166	0.05611	0.04768	0.14584	0.24187	762	0.49152	0.05637	0.04783	0.14587	0.24192
707	0.49192	0.05604	0.04783	0.14621	0.24221	763	0.49257	0.05617	0.04806	0.14585	0.24187
708	0.49176	0.05610	0.04804	0.14589	0.24192	764	0.49168	0.05610	0.04821	0.14572	0.24178
709	0.49158	0.05580	0.04775	0.14637	0.24230	765	0.49161	0.05621	0.04778	0.14593	0.24197
710	0.49067	0.05607	0.04786	0.14611	0.24211	766	0.49116	0.05575	0.04810	0.14622	0.24221
711	0.49142	0.05610	0.04807	0.14590	0.24192	767	0.49106	0.05603	0.04794	0.14608	0.24206
712	0.49080	0.05609	0.04817	0.14574	0.24178	768	0.49241	0.05625	0.04798	0.14585	0.24187
713	0.49246	0.05571	0.04797	0.14579	0.24182	769	0.49100	0.05619	0.04764	0.14575	0.24182
714	0.49218	0.05604	0.04801	0.14592	0.24197	770	0.49083	0.05647	0.04786	0.14574	0.24178
715	0.49091	0.05612	0.04813	0.14580	0.24182	771	0.49165	0.05600	0.04786	0.14570	0.24178
716	0.49153	0.05604	0.04768	0.14607	0.24206	772	0.49171	0.05616	0.04743	0.14595	0.24197
717	0.49146	0.05609	0.04764	0.14633	0.24230	773	0.49191	0.05629	0.04806	0.14565	0.24173
718	0.49193	0.05610	0.04777	0.14601	0.24202	774	0.49138	0.05604	0.04779	0.14604	0.24206
719	0.49191	0.05607	0.04785	0.14613	0.24211	775	0.49009	0.05596	0.04809	0.14578	0.24182
720	0.49130	0.05644	0.04777	0.14584	0.24187	776	0.49149	0.05574	0.04792	0.14640	0.24235
721	0.49105	0.05635	0.04801	0.14567	0.24173	777	0.49122	0.05623	0.04809	0.14561	0.24168
722	0.49161	0.05595	0.04801	0.14593	0.24187	778	0.49227	0.05603	0.04784	0.14619	0.24216
723	0.49228	0.05592	0.04789	0.14620	0.24216	779	0.49182	0.05603	0.04775	0.14591	0.24192
724	0.49289	0.05608	0.04814	0.14584	0.24187	780	0.49203	0.05619	0.04810	0.14579	0.24182
725	0.49058	0.05598	0.04807	0.14600	0.24202	781	0.49163	0.05629	0.04782	0.14592	0.24197
726	0.49229	0.05631	0.04789	0.14589	0.24192	782	0.49155	0.05601	0.04776	0.14606	0.24206
727	0.49146	0.05618	0.04817	0.14570	0.24178	783	0.49217	0.05625	0.04792	0.14591	0.24192
728	0.49190	0.05619	0.04783	0.14595	0.24197	784	0.49135	0.05593	0.04824	0.14543	0.24154

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่สุ่ม แบบเมตริกซ์เชิงพหุคูณ.

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน

No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
785	0.49109	0.05621	0.04765	0.14618	0.24216	841	0.49143	0.05619	0.04790	0.14569	0.24178
786	0.49190	0.05596	0.04797	0.14614	0.24211	842	0.49082	0.05656	0.04736	0.14586	0.24192
787	0.49222	0.05602	0.04789	0.14610	0.24211	843	0.49211	0.05601	0.04802	0.14594	0.24197
788	0.49243	0.05589	0.04787	0.14627	0.24226	844	0.49213	0.05654	0.04767	0.14576	0.24182
789	0.49236	0.05600	0.04781	0.14609	0.24206	845	0.49158	0.05598	0.04835	0.14573	0.24178
790	0.49166	0.05626	0.04809	0.14570	0.24178	846	0.49147	0.05578	0.04702	0.14556	0.24163
791	0.49136	0.05610	0.04800	0.14598	0.24202	847	0.49146	0.05617	0.04797	0.14578	0.24182
792	0.49004	0.05559	0.04784	0.14640	0.24235	848	0.49149	0.05587	0.04796	0.14625	0.24221
793	0.49133	0.05629	0.04752	0.14625	0.24221	849	0.49192	0.05595	0.04793	0.14578	0.24182
794	0.49172	0.05620	0.04786	0.14602	0.24202	850	0.49168	0.05593	0.04809	0.14603	0.24202
795	0.49180	0.05618	0.04784	0.14593	0.24197	851	0.49180	0.05591	0.04801	0.14613	0.24211
796	0.49173	0.05563	0.04817	0.14579	0.24182	852	0.49152	0.05539	0.04853	0.14585	0.24187
797	0.49106	0.05621	0.04785	0.14601	0.24202	853	0.49120	0.05616	0.04801	0.14582	0.24187
798	0.49215	0.05631	0.04773	0.14598	0.24202	854	0.49231	0.05610	0.04783	0.14615	0.24216
799	0.49202	0.05590	0.04808	0.14608	0.24206	855	0.49186	0.05612	0.04793	0.14600	0.24202
800	0.49225	0.05559	0.04828	0.14601	0.24202	856	0.49169	0.05600	0.04773	0.14633	0.24230
801	0.49185	0.05625	0.04820	0.14542	0.24154	857	0.49212	0.05601	0.04802	0.14605	0.24206
802	0.49168	0.05583	0.04783	0.14638	0.24235	858	0.49083	0.05590	0.04785	0.14620	0.24216
803	0.49275	0.05597	0.04792	0.14593	0.24197	859	0.49147	0.05620	0.04779	0.14596	0.24197
804	0.49115	0.05626	0.04774	0.14606	0.24206	860	0.49161	0.05603	0.04779	0.14603	0.24202
805	0.49216	0.05544	0.04793	0.14638	0.24235	861	0.49172	0.05649	0.04747	0.14602	0.24202
806	0.49194	0.05604	0.04801	0.14602	0.24202	862	0.49126	0.05629	0.04765	0.14595	0.24197
807	0.49168	0.05616	0.04819	0.14573	0.24178	863	0.49275	0.05654	0.04774	0.14568	0.24173
808	0.49124	0.05628	0.04790	0.14585	0.24187	864	0.49191	0.05589	0.04791	0.14571	0.24178
809	0.49187	0.05621	0.04762	0.14616	0.24216	865	0.49190	0.05598	0.04799	0.14610	0.24211
810	0.49118	0.05596	0.04750	0.14628	0.24226	866	0.49170	0.05645	0.04734	0.14628	0.24226
811	0.49216	0.05597	0.04809	0.14595	0.24197	867	0.49180	0.05588	0.04826	0.14594	0.24197
812	0.49156	0.05638	0.04780	0.14589	0.24192	868	0.49191	0.05656	0.04786	0.14565	0.24173
813	0.49226	0.05636	0.04781	0.14587	0.24192	869	0.49131	0.05606	0.04785	0.14601	0.24202
814	0.49216	0.05606	0.04804	0.14591	0.24192	870	0.49136	0.05583	0.04817	0.14607	0.24206
815	0.49182	0.05624	0.04766	0.14602	0.24202	871	0.49061	0.05590	0.04785	0.14577	0.24182
816	0.49076	0.05623	0.04790	0.14592	0.24197	872	0.49136	0.05630	0.04780	0.14596	0.24197
817	0.49051	0.05548	0.04815	0.14597	0.24197	873	0.49109	0.05633	0.04751	0.14603	0.24206
818	0.49092	0.05626	0.04757	0.14614	0.24211	874	0.49117	0.05642	0.04746	0.14613	0.24211
819	0.49205	0.05598	0.04841	0.14563	0.24173	875	0.49094	0.05569	0.04801	0.14635	0.24230
820	0.49252	0.05594	0.04800	0.14600	0.24202	876	0.49181	0.05604	0.04807	0.14591	0.24192
821	0.49088	0.05623	0.04798	0.14563	0.24168	877	0.49298	0.05621	0.04816	0.14568	0.24173
822	0.49181	0.05580	0.04798	0.14587	0.24192	878	0.49128	0.05614	0.04794	0.14589	0.24192
823	0.49182	0.05614	0.04748	0.14622	0.24221	879	0.49234	0.05619	0.04783	0.14586	0.24192
824	0.49126	0.05602	0.04795	0.14610	0.24211	880	0.49110	0.05624	0.04831	0.14552	0.24163
825	0.49242	0.05595	0.04803	0.14610	0.24211	881	0.49143	0.05585	0.04789	0.14634	0.24230
826	0.49092	0.05654	0.04795	0.14551	0.24158	882	0.49253	0.05619	0.04710	0.14620	0.24216
827	0.49182	0.05612	0.04795	0.14601	0.24202	883	0.49248	0.05636	0.04781	0.14589	0.24192
828	0.49214	0.05622	0.04821	0.14566	0.24173	884	0.49136	0.05567	0.04813	0.14601	0.24202
829	0.49137	0.05616	0.04770	0.14606	0.24206	885	0.49151	0.05606	0.04809	0.14593	0.24197
830	0.49145	0.05603	0.04793	0.14611	0.24211	886	0.49174	0.05576	0.04842	0.14588	0.24192
831	0.49162	0.05656	0.04779	0.14570	0.24178	887	0.49180	0.05592	0.04800	0.14587	0.24192
832	0.49232	0.05603	0.04757	0.14618	0.24216	888	0.49193	0.05616	0.04809	0.14578	0.24182
833	0.49121	0.05602	0.04818	0.14582	0.24187	889	0.49182	0.05603	0.04796	0.14608	0.24206
834	0.49243	0.05629	0.04797	0.14580	0.24182	890	0.49163	0.05599	0.04816	0.14592	0.24197
835	0.49183	0.05628	0.04754	0.14596	0.24197	891	0.49218	0.05588	0.04808	0.14613	0.24211
836	0.49148	0.05635	0.04786	0.14579	0.24182	892	0.49145	0.05634	0.04770	0.14598	0.24202
837	0.49199	0.05643	0.04775	0.14590	0.24192	893	0.49153	0.05626	0.04802	0.14576	0.24182
838	0.49251	0.05616	0.04797	0.14596	0.24197	894	0.49184	0.05564	0.04838	0.14583	0.24187
839	0.49220	0.05646	0.04761	0.14558	0.24168	895	0.49182	0.05612	0.04768	0.14628	0.24226
840	0.49211	0.05555	0.04798	0.14652	0.24245	896	0.49133	0.05586	0.04808	0.14613	0.24211

ตาราง 27 (ต่อ) แสดงความคลาดเคลื่อนมาตรฐานของค่าคะแนนเฉลี่ยที่ผู้แบบเมตริกซ์เชิงพหุคูณ

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง 600 คน

No.	Mean	VarE	VarI	VarEI	S.E.Mean	No.	Mean	VarE	VarI	VarEI	S.E.Mean
897	0.49149	0.05613	0.04794	0.14600	0.24202	953	0.49219	0.05632	0.04790	0.14582	0.24187
898	0.49049	0.05581	0.04824	0.14598	0.24202	954	0.49105	0.05575	0.04747	0.14629	0.24226
899	0.49103	0.05589	0.04797	0.14618	0.24216	955	0.49157	0.05628	0.04768	0.14577	0.24182
900	0.49223	0.05556	0.04798	0.14613	0.24211	956	0.49179	0.05603	0.04805	0.14579	0.24182
901	0.49069	0.05605	0.04798	0.14603	0.24202	957	0.49141	0.05606	0.04797	0.14556	0.24163
902	0.49112	0.05536	0.04778	0.14620	0.24216	958	0.49142	0.05592	0.04771	0.14590	0.24192
903	0.49105	0.05581	0.04781	0.14587	0.24192	959	0.49107	0.05610	0.04800	0.14588	0.24192
904	0.49207	0.05604	0.04763	0.14607	0.24206	960	0.49170	0.05561	0.04847	0.14591	0.24192
905	0.49204	0.05618	0.04804	0.14572	0.24178	961	0.49199	0.05644	0.04794	0.14566	0.24173
906	0.49170	0.05669	0.04787	0.14545	0.24154	962	0.49142	0.05596	0.04784	0.14624	0.24221
907	0.49092	0.05614	0.04822	0.14557	0.24168	963	0.49077	0.05612	0.04769	0.14623	0.24221
908	0.49290	0.05646	0.04808	0.14555	0.24163	964	0.49235	0.05631	0.04779	0.14590	0.24192
909	0.49156	0.05583	0.04798	0.14620	0.24216	965	0.49085	0.05662	0.04736	0.14582	0.24187
910	0.49104	0.05642	0.04769	0.14595	0.24197	966	0.49098	0.05616	0.04816	0.14548	0.24158
911	0.49202	0.05598	0.04807	0.14597	0.24197	967	0.49016	0.05640	0.04709	0.14583	0.24187
912	0.49218	0.05593	0.04777	0.14626	0.24221	968	0.49122	0.05620	0.04787	0.14600	0.24202
913	0.49022	0.05589	0.04821	0.14591	0.24192	969	0.49210	0.05592	0.04798	0.14618	0.24216
914	0.49167	0.05626	0.04782	0.14561	0.24168	970	0.49176	0.05613	0.04763	0.14638	0.24235
915	0.49180	0.05660	0.04770	0.14576	0.24182	971	0.49178	0.05582	0.04804	0.14620	0.24216
916	0.49083	0.05621	0.04785	0.14552	0.24163	972	0.49210	0.05595	0.04814	0.14600	0.24202
917	0.49110	0.05622	0.04825	0.14556	0.24163	973	0.49168	0.05619	0.04732	0.14601	0.24202
918	0.49145	0.05618	0.04805	0.14584	0.24187	974	0.49141	0.05604	0.04792	0.14611	0.24211
919	0.49144	0.05588	0.04846	0.14572	0.24178	975	0.49178	0.05628	0.04765	0.14587	0.24192
920	0.49138	0.05589	0.04782	0.14596	0.24197	976	0.49111	0.05633	0.04807	0.14562	0.24168
921	0.49140	0.05650	0.04771	0.14586	0.24192	977	0.49092	0.05568	0.04821	0.14606	0.24206
922	0.49167	0.05567	0.04818	0.14623	0.24221	978	0.49140	0.05602	0.04800	0.14604	0.24206
923	0.49164	0.05616	0.04797	0.14582	0.24187	979	0.49163	0.05606	0.04822	0.14575	0.24182
924	0.49131	0.05604	0.04810	0.14579	0.24182	980	0.49176	0.05635	0.04770	0.14601	0.24202
925	0.49144	0.05588	0.04763	0.14630	0.24226	981	0.49085	0.05618	0.04741	0.14617	0.24216
926	0.49153	0.05617	0.04779	0.14576	0.24182	982	0.49139	0.05600	0.04803	0.14590	0.24192
927	0.49168	0.05627	0.04772	0.14608	0.24206	983	0.49218	0.05592	0.04807	0.14593	0.24197
928	0.49239	0.05612	0.04796	0.14583	0.24187	984	0.49171	0.05597	0.04812	0.14595	0.24197
929	0.49168	0.05622	0.04819	0.14563	0.24173	985	0.49181	0.05579	0.04772	0.14608	0.24206
930	0.49099	0.05578	0.04790	0.14567	0.24173	986	0.49145	0.05617	0.04764	0.14626	0.24221
931	0.49079	0.05624	0.04784	0.14593	0.24197	987	0.49165	0.05587	0.04812	0.14598	0.24202
932	0.49166	0.05608	0.04792	0.14608	0.24206	988	0.49009	0.05592	0.04803	0.14609	0.24206
933	0.49163	0.05636	0.04811	0.14561	0.24168	989	0.49213	0.05594	0.04783	0.14624	0.24221
934	0.49331	0.05633	0.04807	0.14569	0.24178	990	0.49121	0.05609	0.04835	0.14564	0.24163
935	0.49131	0.05608	0.04788	0.14600	0.24202	991	0.49190	0.05617	0.04737	0.14588	0.24192
936	0.49095	0.05592	0.04759	0.14602	0.24202	992	0.49201	0.05631	0.04783	0.14586	0.24187
937	0.49249	0.05635	0.04756	0.14614	0.24211	993	0.49139	0.05584	0.04815	0.14587	0.24192
938	0.49173	0.05613	0.04792	0.14616	0.24216	994	0.49150	0.05599	0.04802	0.14588	0.24192
939	0.49168	0.05581	0.04811	0.14615	0.24216	995	0.49164	0.05625	0.04774	0.14599	0.24202
940	0.49195	0.05626	0.04762	0.14589	0.24192	996	0.49170	0.05598	0.04782	0.14611	0.24211
941	0.49117	0.05597	0.04799	0.14610	0.24211	997	0.49257	0.05610	0.04788	0.14597	0.24197
942	0.49209	0.05620	0.04802	0.14585	0.24187	998	0.49181	0.05623	0.04798	0.14566	0.24173
943	0.49094	0.05581	0.04788	0.14632	0.24226	999	0.49131	0.05629	0.04772	0.14591	0.24192
944	0.49160	0.05635	0.04754	0.14619	0.24216	1,000	0.49116	0.05605	0.04767	0.14619	0.24216
945	0.49167	0.05636	0.04807	0.14564	0.24173						
946	0.49112	0.05584	0.04818	0.14589	0.24192						
947	0.49195	0.05613	0.04791	0.14601	0.24202						
948	0.49118	0.05575	0.04768	0.14657	0.24250						
949	0.49207	0.05618	0.04798	0.14590	0.24192						
950	0.49160	0.05581	0.04792	0.14588	0.24192						
951	0.49120	0.05608	0.04816	0.14580	0.24182						
952	0.49187	0.05590	0.04769	0.14616	0.24216						
TOTAL	0.49162	0.05607	0.04790	0.14597	0.24197						

ตาราง 28 เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร
ประชากรทั้งหมด 236,9700 และความแปรปรวนคะแนนของประชากรทั้งหมด 12,919,4717 กรณีข้อสอบมี

ลักษณะการแจกแจงค่าความยากของข้อสอบแบบปกติ จำนวนตัวอย่างผู้ตอบ เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chis-quare	No.	Mean	Variance	t-value	Chis-quare
1	234.3467	12,952.4004	-0.3498	600.5267	57	235.4883	12,850.0146	-0.1038	595.7787
2	240.1900	13,538.3408	0.9094	627.6933	58	237.2483	13,196.9834	0.2755	611.8666
3	231.7250	12,426.1943	-0.9148	576.1296	59	238.7183	13,144.0977	0.5923	609.4146
4	235.4483	13,484.8232	-0.1124	625.2120	60	234.4133	13,145.2979	-0.3955	609.4702
5	236.0933	12,550.2217	0.4576	581.8800	61	233.3300	13,214.0547	-0.5689	612.6581
6	238.9833	13,005.8027	0.6494	603.0027	62	231.0267	12,486.1299	-1.0653	578.9085
7	242.2850	12,651.7588	1.3609	586.5877	63	242.0183	12,951.5859	1.3034	600.4889
8	235.9967	12,172.5674	0.0057	564.3704	64	233.4000	12,791.9736	-0.5538	593.0887
9	233.3750	13,017.0645	-0.5592	603.5248	65	232.8700	13,426.6611	-0.6681	622.5154
10	241.4167	13,187.9531	1.1738	611.4479	66	236.6600	12,774.4951	0.1487	592.2783
11	237.2517	13,016.5611	0.2762	603.5024	67	232.6050	13,212.2881	-0.7252	612.5762
12	236.9917	12,987.8760	0.2202	602.1715	68	240.3700	13,246.3135	0.9482	614.1537
13	234.7017	12,920.8291	-0.2733	599.0629	69	238.8217	13,032.9932	0.6145	604.2633
14	232.2900	12,783.1543	-0.7931	592.6798	70	229.2400	12,263.8760	-1.4503	568.6039
15	234.7100	12,596.1631	-0.2715	584.0101	71	238.1233	13,162.6826	0.4641	610.2763
16	231.4083	13,430.7607	-0.9831	622.7055	72	241.1450	12,492.2432	1.1152	579.1919
17	234.2617	13,368.4727	-0.3661	619.8175	73	241.4183	12,002.4590	1.1741	556.4835
18	238.8650	13,476.9004	0.6239	624.8447	74	237.7267	12,998.5293	0.3786	602.6654
19	237.1533	13,084.5039	0.2550	606.6516	75	236.6583	13,005.3223	0.1483	602.9804
20	243.5483	12,891.0547	1.6332	597.6825	76	237.3483	13,879.7080	0.2970	643.5205
21	235.3867	13,080.7588	-0.1257	606.4779	77	235.0250	12,825.9795	-0.2037	594.6653
22	231.8783	12,216.3525	-0.8818	566.4005	78	227.5883	12,873.4199	-1.8063	596.8649
23	233.1767	13,198.7930	-0.6020	611.9505	79	235.1867	13,188.8604	-0.1688	611.4900
24	246.1300	13,194.4873	2.1895*	611.7509	80	240.9450	12,722.0771	1.0721	589.8480
25	235.1367	13,296.0146	-0.1796	616.4581	81	230.0317	11,931.7920	-1.2797	553.2071
26	235.7583	13,365.6240	-0.0456	619.6855	82	236.1533	13,139.3018	0.0395	609.1822
27	234.8583	13,324.1631	-0.2396	617.7632	83	235.1933	13,113.2646	-0.1674	607.9850
28	236.4117	12,463.8105	0.0952	577.8737	84	234.6417	12,827.5811	-0.2863	594.7396
29	237.2117	13,100.1719	0.2676	607.3780	85	236.6067	13,339.2236	0.1372	618.4614
30	236.0250	12,877.5752	0.0119	597.0575	86	237.1000	12,908.4238	0.2435	598.4878
31	224.3550	13,149.7100	-2.5031	609.6748	87	230.6383	12,507.3984	-1.1490	579.8946
32	235.1150	13,000.4443	-0.1843	602.7542	88	242.1833	13,474.8379	1.3390	624.7491
33	237.4767	13,113.9766	0.3247	608.0180	89	232.2200	12,641.1602	-0.8081	586.0963
34	234.5100	13,502.3770	-0.3146	626.0259	90	232.5050	13,392.3252	-0.7467	620.9234
35	234.9267	12,806.8594	-0.2248	593.7788	91	238.9867	12,961.3291	0.6501	600.9407
36	232.8700	12,671.3154	-0.6681	587.4945	92	237.1717	12,232.3281	0.2590	567.1412
37	239.5033	13,230.8916	0.7614	613.4387	93	236.3083	12,235.5713	0.0729	567.2916
38	233.6333	12,979.1777	-0.5036	601.7682	94	233.5783	12,654.3330	-0.5154	586.7071
39	237.6750	14,012.1611	0.3674	649.6616	95	235.1950	12,069.8086	-0.1670	559.6061
40	234.0600	13,739.5762	-0.4116	637.0234	96	234.9150	12,913.6055	-0.2274	598.7280
41	240.2283	12,803.1260	0.9177	593.6057	97	235.6583	12,848.7383	-0.0672	595.7205
42	236.9850	12,488.7373	0.2187	579.0294	98	231.9083	12,581.1465	-0.8753	583.3138
43	238.5450	12,778.4805	0.5549	592.4631	99	241.3417	12,865.1621	1.1576	596.4820
44	239.8417	12,547.3408	0.8344	581.7465	100	230.9217	13,187.4160	-1.0879	611.4230
45	226.7183	12,730.2344	-1.9938	590.2262	101	233.2633	13,235.0420	-0.5833	613.6311
46	233.1133	13,144.3809	-0.6156	609.4277	102	233.6083	12,619.9531	-0.5089	585.1131
47	232.7900	13,102.3936	-0.6853	607.4810	103	237.3733	13,698.9727	0.3024	635.1409
48	236.7983	12,926.6338	0.1785	599.3321	104	235.0000	13,112.6006	-0.2090	607.9543
49	238.5000	12,772.4912	0.5452	592.1854	105	239.4167	12,812.2002	0.7428	594.0265
50	234.0033	12,608.1738	-0.4238	584.5669	106	233.4200	12,964.8652	-0.5495	601.1046
51	239.8950	12,630.3525	0.8458	585.5952	107	235.1400	13,822.0107	-0.1789	640.8454
52	239.5417	13,020.4844	0.7697	603.6834	108	232.2683	12,448.6963	-0.7977	577.1729
53	236.7683	13,389.8799	0.1720	620.8101	109	236.8000	12,341.6631	0.1789	572.2104
54	235.5900	13,038.8770	-0.0819	604.5361	110	237.0650	12,813.5283	0.2360	594.0880
55	236.4950	12,913.2100	0.1131	598.7097	111	239.4717	13,389.6172	0.7546	620.7979
56	238.1617	13,277.4365	0.4723	615.5967	112	231.2400	13,555.9492	-1.0193	628.5097

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร
ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีการแจกแจง
แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
113	239.7717	13,289.1836	0.8193	616.1414	169	240.0783	13,745.0820	0.8854	637.2787
114	233.2283	12,543.1494	-0.5908	581.5521	170	234.7217	12,555.6455	-0.2690	582.1315
115	239.6533	12,618.1572	0.7938	585.0298	171	239.0850	12,529.5449	0.6713	580.9214
116	234.2483	13,299.4639	-0.3710	616.6180	172	236.3767	13,339.0762	0.0876	618.4546
117	235.9200	12,340.3340	-0.0108	572.1488	173	241.0750	13,352.3984	1.1001	619.0723
118	233.3517	13,153.7959	-0.5643	609.8642	174	235.0283	13,200.7568	-0.2029	612.0415
119	238.7550	13,036.3105	0.6002	604.4171	175	239.7367	13,901.4531	0.8117	644.5287
120	234.3200	13,133.6934	-0.3556	608.9322	176	235.3883	13,093.9160	-0.1254	607.0880
121	237.5733	12,765.7813	0.3455	591.8743	177	240.5550	13,194.5293	0.9881	611.7528
122	237.5900	12,651.3379	0.3491	586.5682	178	230.3267	13,137.0947	-1.2162	609.0899
123	237.6300	12,987.1416	0.3577	602.1375	179	236.0867	12,967.5381	0.0251	601.2286
124	239.1100	13,284.0410	0.6767	615.9029	180	235.1967	12,784.7764	-0.1667	592.7550
125	236.2733	13,285.8818	0.0654	615.9883	181	240.9467	13,709.3125	1.0725	635.6203
126	236.2717	13,194.5303	0.0650	611.7529	182	233.7133	12,755.1836	-0.4863	591.3829
127	231.1333	12,750.6602	-1.0423	591.1732	183	231.3367	13,427.6924	-0.9985	622.5632
128	234.8783	12,098.3594	-0.2353	560.9298	184	231.8350	12,890.1631	-0.8911	597.6411
129	236.8733	13,192.8057	0.1947	611.6729	185	236.0217	12,849.0215	0.0111	595.7336
130	234.5883	12,431.8135	-0.2978	576.3902	186	237.6900	13,549.1055	0.3707	628.1924
131	238.5467	13,557.5439	0.5553	628.5837	187	237.4083	12,039.1553	0.3100	558.1849
132	236.3183	12,910.1426	0.0751	598.5675	188	236.9183	12,933.4658	0.2044	599.6488
133	237.3633	12,705.4902	0.3003	589.0789	189	239.5650	12,746.3418	0.7747	590.9730
134	239.8883	13,275.0059	0.8444	615.4840	190	234.2283	13,019.6006	-0.3753	603.6424
135	236.7400	13,379.0439	0.1659	620.3077	191	234.9100	12,268.7930	-0.2264	568.8319
136	237.0617	12,540.9111	0.2396	581.4484	192	234.7233	13,861.6328	-0.2687	619.5004
137	235.8983	13,272.4639	-0.0154	615.3662	193	237.6133	12,164.1904	0.3541	563.9820
138	238.9500	12,785.5537	0.6422	592.7910	194	234.1783	13,008.6025	-0.3861	603.1325
139	237.2617	13,185.9180	0.2784	611.3535	195	236.1267	12,609.7598	0.0338	584.6405
140	238.6250	13,379.4287	0.5722	620.3255	196	237.8567	13,395.9863	0.4066	621.0932
141	232.5150	12,199.3066	-0.7446	565.6102	197	238.3017	12,456.6504	0.5025	577.5417
142	238.8767	13,005.9277	0.6264	603.0085	198	235.2167	12,739.2246	-0.1623	590.6430
143	232.8333	13,551.9121	-0.6760	628.3225	199	241.3133	12,631.7480	1.1515	585.6599
144	235.6000	12,924.4473	-0.5107	599.2307	200	241.8733	13,324.9590	1.2722	617.8001
145	235.6733	12,204.8418	-0.0639	565.8668	201	237.1100	12,821.1963	0.2457	594.4435
146	236.2150	13,058.1240	0.0528	605.4285	202	235.8117	13,139.7041	-0.0341	609.2109
147	236.7950	12,403.0195	0.6088	575.0551	203	232.1483	12,923.3906	-0.8236	599.1817
148	234.5950	13,240.9209	-0.2963	613.9037	204	237.9383	12,456.1094	0.4242	577.5166
149	235.5650	13,084.7783	-0.0873	606.6643	205	235.9067	13,368.3516	-0.0136	619.8119
150	238.2883	13,395.2285	0.4996	621.0580	206	237.4067	12,547.1670	0.3096	581.7384
151	234.5617	13,777.6738	-0.3035	638.7898	207	241.2250	12,958.5234	1.1325	600.8106
152	237.4133	13,130.3193	0.3110	608.7758	208	238.7350	12,386.5908	0.5959	574.2934
153	235.8150	12,676.3193	-0.0834	587.7265	209	239.6667	12,337.1445	0.7966	572.0009
154	233.9967	12,955.9199	-0.4253	600.6899	210	234.5333	12,174.5664	-0.3096	564.4631
155	239.3750	12,672.4102	0.7338	587.5452	211	231.7783	12,593.4736	-0.9033	583.8654
156	234.4283	12,814.5605	-0.3322	594.1359	212	235.5950	13,514.4600	-0.0808	626.5861
157	237.2617	12,483.1064	0.2784	578.7683	213	236.2233	13,341.4326	0.0546	618.5638
158	237.3533	12,407.8682	0.2981	575.2800	214	232.8400	12,988.2744	-0.6745	602.1900
159	241.8917	12,807.1201	1.2761	593.7909	215	238.8550	12,655.7422	0.6217	586.7724
160	237.3800	12,872.0420	0.3039	596.8010	216	238.0517	13,068.5078	-0.6289	605.9099
161	234.8017	13,098.9492	-0.2518	607.3213	217	240.4100	12,676.0654	0.9568	587.7147
162	236.3767	12,251.5371	0.0876	568.0318	218	232.5917	12,352.3643	-0.7280	572.7066
163	228.7650	13,024.9766	-1.5527	603.8916	219	231.3567	12,593.0947	-0.9942	583.8678
164	235.4917	12,950.1924	-0.1031	600.4243	220	241.5767	13,394.7119	1.2083	621.0341
165	230.3750	12,923.3320	-1.2057	599.1790	221	232.6717	13,224.2891	-0.7108	613.1326
166	235.4583	13,001.9033	-0.1103	602.8219	222	240.1400	14,215.7998	0.8986	659.1031
167	235.8333	13,362.5234	-0.0295	619.5417	223	233.9117	12,741.4277	-0.4436	590.7451
168	235.6283	12,223.0332	-0.0736	566.7102	224	234.3883	12,939.1143	-0.3409	599.9107

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร

ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีนี้ข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
225	241.2650	13,247.1514	1.1411	614.1926	281	239.1500	12,989.0723	0.6853	602.2270
226	231.1600	13,051.6006	-1.0366	605.1260	282	232.0767	13,471.4268	-0.8390	624.5909
227	233.9583	12,779.2842	-0.4335	592.5003	283	238.9900	13,238.0430	0.6508	613.7703
228	229.7633	13,535.9502	-1.3376	627.5825	284	231.1267	12,200.1240	-1.0438	565.6481
229	230.8933	13,287.1006	-1.0940	616.0448	285	239.1500	13,522.4482	0.6853	626.9565
230	237.3717	13,158.4756	0.3021	610.0812	286	235.6783	13,018.4941	-0.0629	603.5911
231	233.2167	12,706.2500	-0.5934	589.1142	287	231.7233	13,015.5098	-0.9152	603.4527
232	236.1150	12,096.2334	0.0312	560.8313	288	237.5983	12,525.4277	0.3509	580.7305
233	238.4933	13,085.3721	0.5438	606.6918	289	237.9383	12,951.6621	0.4242	600.4925
234	229.9050	12,409.8008	-1.3070	575.3696	290	233.1083	12,878.5088	-0.6167	597.1008
235	239.1867	12,354.4432	0.6932	572.8032	291	229.8117	12,771.4902	-1.3271	592.1390
236	237.0183	12,555.4219	0.2259	582.1211	292	236.1767	12,665.9053	0.0445	587.2436
237	230.8233	12,637.1240	-1.1091	585.9092	293	237.1850	13,222.9609	0.2618	613.0710
238	239.7983	12,997.1348	0.8250	602.6036	294	232.9433	13,210.1133	-0.6523	612.4753
239	239.0167	13,345.9297	0.6566	618.7724	295	235.0433	12,873.7813	-0.1997	596.8816
240	236.3483	13,222.1621	0.0815	613.0340	296	236.2150	12,693.6563	0.0528	588.5303
241	237.5933	13,111.5137	0.3498	607.9039	297	245.0400	12,420.8398	1.9546	575.8814
242	239.3833	12,939.7129	0.7356	599.9385	298	239.5017	12,820.8535	0.7611	594.4277
243	231.9450	12,756.9316	-0.8674	591.4640	299	236.4267	12,894.1553	0.0984	597.8262
244	236.3617	12,828.3301	0.0844	594.7743	300	232.8167	13,003.5527	-0.6796	602.8983
245	238.1533	12,591.3691	0.4705	583.7878	301	234.5717	12,472.2266	-0.3013	578.2639
246	233.2767	12,709.0096	-0.5804	589.1958	302	233.3067	12,746.5439	-0.5740	590.9824
247	234.5333	13,229.4648	-0.3096	613.3726	303	237.5417	13,166.3008	0.3387	610.4440
248	231.4017	12,750.4395	-0.9645	591.1630	304	237.8133	12,733.3008	0.3972	590.3683
249	229.0717	12,526.8291	-1.4886	590.7955	305	237.9367	13,214.5000	0.4238	612.6787
250	235.7800	13,003.3770	-0.0409	602.8902	306	237.2700	13,822.6182	0.2802	640.8736
251	233.4900	13,227.1250	-0.5344	613.2641	307	242.1067	13,580.4893	1.3225	629.6475
252	234.8000	12,620.2871	-0.2521	585.1286	308	236.3300	13,386.1543	0.0776	620.6373
253	236.8283	13,708.7686	0.1850	635.5951	309	226.1767	12,872.4834	-2.1105	596.8214
254	231.7417	13,225.6289	-0.9112	613.1947	310	233.3350	13,749.5469	-0.5678	637.4857
255	238.6300	13,534.4941	0.5732	627.5150	311	235.6400	12,552.3447	-0.0711	581.9785
256	235.6650	12,841.8711	-0.0657	595.4021	312	229.5550	12,443.9951	-1.3825	576.9549
257	237.8783	13,334.5557	0.4112	618.2450	313	239.5217	13,612.9463	0.7654	631.1523
258	236.1550	12,797.3652	0.0399	593.3386	314	237.4067	13,124.6826	0.3096	608.5144
259	236.6567	13,090.0586	0.1480	606.9031	315	238.7867	12,857.6104	0.6070	596.1319
260	240.2367	12,507.2695	0.9195	579.8886	316	230.9317	13,353.0000	-1.0858	619.1002
261	233.6950	13,716.9219	-0.4903	635.9731	317	240.2733	13,063.6377	0.9274	605.6841
262	241.4483	13,439.0938	1.1806	623.0918	318	239.2017	12,648.5303	0.6964	586.4380
263	236.7167	13,061.1084	0.1609	605.5669	319	234.7367	13,021.5332	-0.2658	603.7320
264	243.6350	12,808.9082	1.6518	593.3738	320	235.8083	13,205.2256	-0.0348	612.2487
265	236.1883	13,305.3271	0.4781	616.8899	321	234.9417	12,708.8945	-0.2216	589.2368
266	234.0467	12,959.0430	-0.4145	600.8347	322	235.9633	13,165.7949	-0.0014	610.4206
267	231.4283	12,618.2432	-0.9787	585.0338	323	227.1617	13,560.6445	-1.8982	628.7274
268	232.3417	12,765.6924	-0.7819	591.8702	324	235.7167	13,144.6338	-0.0546	609.4394
269	240.0233	12,780.0791	0.8735	592.5372	325	236.8967	13,326.6074	0.1997	617.8765
270	237.4950	13,451.8584	0.3286	623.6836	326	236.6117	12,906.0957	0.1383	598.3798
271	232.1883	12,744.3389	-0.8150	590.8901	327	236.8867	13,069.8105	0.1975	605.9703
272	234.2917	12,639.0938	-0.3617	586.0005	328	230.3300	12,676.7793	-1.2154	587.7478
273	230.7900	13,393.7051	-1.1163	620.9874	329	237.3950	12,843.6533	0.3071	595.4847
274	231.7050	12,722.0361	-0.9191	589.8461	330	238.0767	12,907.9609	0.4540	598.4663
275	231.1387	13,236.7422	-1.0416	613.7100	331	241.7217	12,677.2783	1.2395	587.7709
276	231.0550	13,398.1240	-1.0592	621.1923	332	236.4250	12,606.3027	0.0981	584.4802
277	235.1050	12,765.9424	-0.1864	591.8817	333	230.2567	12,878.5518	-1.2312	597.1028
278	234.2867	12,880.8994	-0.3628	597.2116	334	235.4733	12,709.4316	-0.1070	589.2617
279	232.6317	13,253.1631	-0.7194	614.4713	335	238.0150	13,179.6729	0.4407	611.0640
280	242.4517	12,943.4082	1.3968	600.1098	336	239.3233	13,630.8467	0.7227	631.9823

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร

ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chis-square	No.	Mean	Variance	t-value	Chis-square
337	232.3350	12,273.0225	-0.7834	569.0279	393	234.3783	12,777.3994	-0.3430	592.4129
338	239.0267	13,739.7988	0.8587	637.0338	394	237.4600	12,673.1641	0.3211	587.5802
339	232.8233	12,441.4043	-0.6781	576.8348	395	238.1183	12,297.8623	0.4630	570.1796
340	238.3433	12,694.9336	0.5115	586.5895	396	235.4567	12,394.2754	-0.1106	574.6487
341	229.9067	13,043.0029	-1.3067	604.7274	397	238.3167	12,420.5469	0.5057	575.8678
342	242.0567	13,226.7148	1.3117	613.2451	398	232.3000	13,342.2539	-0.7909	618.6019
343	230.6950	13,014.4746	-1.1368	603.4047	399	235.5417	12,876.9854	-0.0923	597.0302
344	237.6433	12,823.9131	0.3606	594.5695	400	242.0533	12,386.5146	1.3110	574.2899
345	229.7783	12,197.2266	-1.3343	565.5137	401	238.1583	13,179.4072	0.4716	611.0517
346	236.5767	12,669.0293	0.1307	587.3885	402	233.1517	12,735.4990	-0.6074	590.4703
347	233.2117	13,161.7148	-0.5944	610.2314	403	238.0733	13,630.3447	0.4533	631.9590
348	238.3950	12,667.1357	0.5226	587.9007	404	240.2617	12,617.3867	0.9249	584.9941
349	238.3917	12,785.5557	0.5219	592.7911	405	238.1850	12,267.1172	0.4773	568.7542
350	236.5233	13,387.1113	0.1192	620.6817	406	240.3917	12,961.3223	0.9529	600.9404
351	237.2733	12,029.3877	0.2809	557.7320	407	235.4133	12,841.9121	-0.1200	595.4040
352	240.6300	12,362.6074	1.0042	573.1815	408	236.7567	13,169.0156	0.1695	610.5699
353	231.8200	12,738.3311	-0.8943	590.6016	409	240.6367	12,539.2871	1.0057	581.3731
354	233.4167	13,181.8428	-0.5503	611.1646	410	235.7233	12,587.5029	-0.0532	583.6086
355	232.7050	13,128.0498	-0.7086	608.6705	411	235.1783	13,701.8750	-0.1706	635.2754
356	239.6050	13,389.6270	0.7834	620.7988	412	237.1000	12,899.0547	0.2435	598.0552
357	228.7417	12,609.8330	-1.5577	584.6439	413	234.6267	13,183.6736	-0.2695	611.2495
358	235.9850	12,685.7490	0.0032	588.1637	414	239.2900	13,201.6650	0.7155	612.0836
359	236.4633	13,430.9639	0.5373	622.7149	415	240.0367	12,936.9473	0.8764	599.6102
360	234.0283	13,003.5986	-0.4184	602.9005	416	232.0100	13,124.5879	-0.8534	608.5100
361	235.7050	13,070.2363	-0.0571	605.9901	417	240.2717	12,121.9893	0.9270	562.0254
362	231.5667	13,115.1572	-0.9489	608.0728	418	234.8000	12,704.2471	-0.2521	598.0213
363	234.8133	12,655.6982	-0.2493	586.7704	419	244.8650	12,690.0088	1.9169	588.3612
364	236.3933	12,995.9521	0.0912	602.5460	420	234.2967	12,396.4463	-0.3606	574.7504
365	231.9000	12,298.1533	-0.8771	570.1931	421	236.3433	12,864.0889	0.0805	596.4322
366	230.0950	13,479.3232	-1.2661	624.9570	422	232.4617	13,285.6826	-0.7561	615.9790
367	235.8867	12,421.0625	-0.0180	575.8917	423	235.7283	13,604.9414	-0.0521	630.7812
368	238.2817	12,581.3965	0.4982	583.3254	424	233.9550	12,009.8340	-0.4342	556.8254
369	238.9267	13,304.5352	0.6372	616.8531	425	235.4583	12,917.4688	-0.1103	598.9071
370	238.5733	13,272.9932	0.5610	615.3907	426	230.2750	12,250.2744	-1.2273	567.9733
371	234.2250	13,224.2295	-0.3761	613.1298	427	234.4117	13,106.4951	-0.3358	607.6712
372	237.1533	12,493.2656	0.2550	579.2393	428	240.6150	12,685.0371	1.0010	588.1306
373	233.8900	13,248.7158	-0.4482	614.2651	429	238.5817	12,492.2090	0.5628	579.1903
374	235.0367	12,400.6064	-0.2011	574.9433	430	240.0033	13,347.2520	0.8692	618.8337
375	231.1533	12,527.0684	-1.0380	580.8066	431	232.0383	13,158.4229	-0.8473	610.0786
376	232.0900	12,739.7051	-0.8362	590.6653	432	238.5900	12,691.6982	0.5646	586.4395
377	235.4050	12,962.2197	-0.1218	600.5183	433	232.6550	12,343.7705	-0.7144	572.3081
378	234.3417	12,898.8145	-0.3509	598.0422	434	237.9667	13,154.9307	0.4303	609.9168
379	236.9067	13,665.4336	0.2019	633.5859	435	235.6317	12,532.1143	-0.0945	581.0405
380	232.0883	13,573.5352	-0.8365	629.3251	436	237.6600	12,998.2148	0.3642	602.6509
381	235.2667	12,972.5293	-0.1516	601.4600	437	234.8283	12,792.1807	-0.2460	593.0883
382	233.0933	13,184.8799	-0.6193	611.3054	438	238.0600	12,732.9375	0.4504	590.3515
383	241.4550	12,373.6260	1.1820	573.6923	439	234.9350	12,643.0244	-0.2230	586.1828
384	239.1800	13,267.9346	0.6918	615.1562	440	234.4050	13,502.4004	-0.3373	626.0270
385	230.5317	13,178.7754	-1.1720	611.0224	441	234.6533	12,900.8682	-0.2837	598.1375
386	232.8750	12,839.7275	-0.6670	595.3027	442	237.2283	13,163.2637	0.2712	610.3032
387	240.3567	12,575.7520	0.9453	583.0637	443	237.3300	12,647.2363	0.2931	596.3780
388	233.1433	12,370.0361	-0.6092	573.5259	444	233.8167	11,854.1230	-0.4641	549.6060
389	235.3567	13,226.9375	-0.1322	613.2554	445	236.0483	13,385.5674	0.0169	620.6101
390	238.0150	13,313.9802	0.4407	617.2915	446	237.8617	12,220.1143	0.4077	566.5749
391	239.1100	13,089.0498	0.6767	606.8623	447	237.1317	12,755.4980	0.2503	591.8975
392	237.4217	13,685.4346	0.3128	634.5132	448	234.4333	13,054.7764	-0.3312	605.2733

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร:

ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
449	234.4183	13,218.5693	-0.3344	612.8674	505	234.4017	12,164.0586	-0.3380	563.9759
450	234.7583	11,584.7061	-0.2611	537.1148	506	239.1267	13,089.5527	0.6803	606.8857
451	240.9433	13,403.2021	1.0718	621.4277	507	237.1700	12,994.5088	0.2586	602.4790
452	236.8200	13,362.8125	0.1832	619.5551	508	237.6033	12,832.6035	0.3520	594.9724
453	235.7883	13,095.2275	-0.0392	607.1488	509	236.3933	12,678.1504	0.0891	587.8113
454	232.8967	12,341.0645	-0.6623	572.1827	510	237.8617	13,045.8037	0.4077	604.8573
455	235.4850	12,961.7246	-0.1045	600.9590	511	234.5750	13,198.7139	-0.3006	611.9468
456	241.4533	13,198.5322	1.1817	611.9384	512	235.5267	13,194.1396	-0.0955	611.7347
457	237.2400	12,979.6816	0.2737	601.7916	513	234.3817	12,651.7744	-0.3423	586.5885
458	237.5317	13,116.2813	0.3365	608.1249	514	232.9767	13,903.8662	-0.6451	644.6406
459	237.6267	12,972.5615	0.3570	601.4615	515	233.0200	12,513.5488	-0.6357	580.1797
460	238.1883	13,020.1982	0.4781	603.6701	516	236.4300	13,311.4141	0.0991	617.1721
461	236.9600	12,792.7598	0.2133	593.1251	517	232.0167	12,899.5527	-0.8520	598.0765
462	233.7000	13,985.0449	0.5883	648.4044	518	240.0833	13,812.7480	0.8864	640.4160
463	240.0950	13,305.9746	0.8889	616.9199	519	235.4617	13,011.5029	-0.1095	603.2669
464	240.1200	12,874.5830	0.8943	596.9188	520	225.9450	11,915.5830	-2.1604	552.4556
465	233.6433	12,556.0264	-0.5014	582.1492	521	243.0500	13,566.0879	1.5258	628.9796
466	234.7150	12,761.8184	-0.2705	591.6905	522	239.9350	13,773.7920	0.8546	638.6086
467	233.3817	12,616.7793	-0.5578	584.9659	523	234.7167	13,348.6406	-0.2701	618.8980
468	236.5783	12,799.5381	0.1311	593.4394	524	237.0433	13,556.3584	0.2313	628.5287
469	232.9533	13,160.6387	-0.6501	610.1815	525	239.0733	12,372.0215	0.6688	573.6180
470	233.4117	13,220.7119	-0.5513	613.4304	526	231.9733	12,970.0957	-0.8613	601.3471
471	236.2783	12,681.7793	0.0664	587.9796	527	230.0750	13,295.5205	-1.2704	616.4352
472	233.5917	13,402.8027	-0.5319	621.4092	528	233.7683	12,756.2691	-0.4702	591.4342
473	238.8833	13,256.7275	0.6278	614.6366	529	236.3533	12,253.0205	0.0826	588.5642
474	240.1167	13,012.3535	0.8936	603.3064	530	236.5217	12,632.0479	0.5499	585.6738
475	236.4083	13,422.0967	0.0945	622.3038	531	236.3633	13,278.1387	0.5158	615.6293
476	237.3283	12,924.8369	0.2927	599.2488	532	243.1167	12,796.7539	1.5401	593.3103
477	233.5083	13,685.3135	-0.5305	634.5076	533	235.8817	12,863.2979	-0.0190	598.3955
478	232.6850	14,187.1689	-0.7079	657.7757	534	238.0767	12,744.5820	0.4540	590.8914
479	240.0367	13,363.8848	0.8764	619.6048	535	238.2900	12,880.7910	0.5000	597.2066
480	239.5500	13,579.6367	0.7715	629.6080	536	236.7300	12,972.1943	0.1638	601.4444
481	240.5017	13,800.7100	0.9766	639.8578	537	235.8783	13,176.2188	-0.0198	610.9039
482	235.8050	12,678.4590	-0.0356	587.8257	538	234.6183	12,794.4414	-0.2913	593.2031
483	237.0767	12,873.8506	0.2385	596.8848	539	240.6850	13,121.1729	1.0161	608.3517
484	236.6817	13,307.1006	0.1534	616.9721	540	231.0350	12,453.9785	-1.0635	577.4178
485	237.3550	12,990.7754	0.2985	602.3059	541	235.6567	12,692.8770	-0.0575	597.7670
486	235.8683	12,511.3281	-0.0219	580.0768	542	232.3350	13,711.3799	-0.7834	635.7161
487	239.7050	12,699.2586	0.8049	588.7900	543	232.3850	12,977.8486	-0.7726	601.7066
488	235.1517	12,523.5127	-0.1764	580.6417	544	235.2067	13,037.3594	-0.1645	604.4658
489	237.2433	13,521.1826	0.2744	626.8978	545	234.7200	12,981.0469	-0.2694	599.5367
490	232.8667	12,950.0986	-0.6688	600.4200	546	238.9317	12,792.2256	0.6382	593.1003
491	232.6600	12,649.0332	-0.7133	586.4614	547	232.7350	13,142.3438	-0.6972	609.3333
492	236.9800	12,588.8682	0.2177	583.6719	548	232.8600	13,553.0752	-0.6702	628.3765
493	238.2367	12,591.3232	0.4885	583.7857	549	240.9217	12,348.1240	1.0671	572.5100
494	231.2233	12,101.5723	-1.0229	561.0788	550	241.8383	14,040.7480	1.2646	650.9870
495	236.9417	12,594.1299	0.2094	583.9158	551	238.4600	12,828.1885	0.5366	594.7677
496	242.1850	12,945.1348	1.3394	600.1898	552	232.8550	12,568.5498	-0.6713	582.7298
497	232.1733	13,267.6826	-0.8182	615.1445	553	232.9733	12,668.2334	-0.6458	587.3516
498	231.3817	13,631.5566	-0.9888	632.0152	554	235.9900	13,026.5313	0.0043	603.9637
499	239.3100	13,264.4785	0.7198	614.9959	555	234.8833	13,420.1934	-0.2342	622.2155
500	239.8633	13,028.5420	0.8390	604.0569	556	232.7467	12,674.6270	-0.6946	587.6480
501	240.2133	12,970.7119	0.9145	601.3757	557	233.0067	12,656.6943	-0.6386	586.8166
502	237.3217	12,585.8896	0.2913	583.5338	558	235.5117	13,032.4961	-0.0988	604.2403
503	237.5833	13,049.1279	0.3477	605.0114	559	241.8533	13,281.3203	1.2679	615.7768
504	234.8983	12,937.4951	-0.2309	599.8356	560	241.8367	13,377.4824	1.2643	620.2353

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร

ประชากรทั้งหมด 235 9700 และความแปรปรวนของประชากรทั้งหมด 12,919.4717 กรณีทดสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
561	241.7733	13,310.0918	1.2506	617.1108	617	234.9067	13,062.1084	-0.2292	605.6132
562	240.9033	12,942.8994	1.0416	600.0862	618	239.1650	13,008.6709	0.6885	603.1356
563	232.9450	12,114.5645	-0.6519	561.6812	619	239.1150	12,652.9248	0.6778	586.6418
564	237.3850	12,826.2119	0.3049	594.6761	620	232.4367	13,106.2461	-0.7614	607.6596
565	234.4350	12,211.0557	-0.3308	566.1549	621	236.0117	12,827.2051	0.0090	594.7221
566	234.4050	12,722.5537	-0.3373	589.8701	622	237.1650	13,233.2178	0.2575	613.5466
567	234.6150	12,989.6592	-0.2920	602.2541	623	235.3217	12,706.2305	-0.1397	589.1133
568	242.0600	13,937.7568	1.3167	646.2119	624	238.1017	13,487.2549	0.4594	625.3248
569	237.4167	12,292.4336	0.3118	569.9279	625	237.0567	12,232.4238	0.2342	567.1456
570	238.3233	13,063.3076	0.5072	605.6688	626	233.4283	13,398.6113	-0.5477	621.2149
571	237.1900	12,760.7588	0.2629	592.6687	627	237.5417	13,214.0732	0.3387	612.6589
572	230.6100	12,477.6777	-1.1551	578.6166	628	235.0783	12,730.3740	-0.1922	590.2327
573	232.6050	12,832.5146	-0.7252	594.9693	629	237.4283	11,888.3740	0.3143	551.1941
574	241.1800	13,244.4150	1.1226	614.0657	630	230.6983	13,314.7832	-1.1361	617.3283
575	234.9367	13,085.8193	-0.2227	605.7853	631	235.9950	13,197.8760	0.0054	611.9080
576	237.8600	13,181.3359	0.4073	611.1411	632	237.9167	13,231.8262	0.4195	613.4820
577	234.3733	12,666.3076	-0.3441	586.2823	633	236.3733	12,958.1172	0.0869	600.7918
578	242.1917	12,512.2900	1.3408	580.1214	634	239.8383	12,810.6650	0.8336	593.9553
579	229.4767	13,140.0127	-1.3993	609.2252	635	233.3050	13,256.2578	-0.5743	614.6148
580	236.2217	13,511.1582	0.0564	626.4330	636	239.7517	13,128.1787	0.8150	608.6765
581	234.8450	12,899.8594	-0.2424	598.0907	637	231.0433	13,530.8359	-1.0617	627.3454
582	237.8267	13,207.5156	0.4001	612.3549	638	239.1783	13,222.6826	0.6314	613.0581
583	237.0533	12,551.4365	0.2335	581.9364	639	230.8783	12,498.8262	-1.0973	579.4971
584	234.0517	13,255.1025	-0.4134	614.5612	640	239.2383	12,917.1221	0.7043	598.8911
585	236.5167	13,132.9414	0.1178	608.8973	641	238.4900	13,091.4678	0.5301	606.9744
586	237.2483	12,564.0215	0.2755	582.5199	642	232.8783	13,181.7715	-0.6663	611.1613
587	237.2317	13,429.7588	0.2719	622.6590	643	237.6950	12,534.5645	0.3717	581.1541
588	239.6167	13,663.8662	0.7859	633.5132	644	236.5817	13,126.8135	0.5628	608.6132
589	238.6967	12,945.9775	0.5876	600.2289	645	234.5617	12,576.1914	-0.3035	583.0341
590	241.4950	12,727.1504	1.1907	590.0832	646	231.7483	12,771.5430	-0.9098	592.1414
591	239.7933	12,866.0273	0.8239	596.5221	647	236.6833	13,545.2021	0.1494	628.0114
592	234.8517	12,520.8760	-0.2410	580.5195	648	237.5417	12,976.4307	0.3387	601.6409
593	235.4163	12,818.5322	-0.1189	594.3200	649	236.9517	13,009.3428	0.6426	603.1668
594	237.0067	13,194.8711	0.2234	611.7687	650	237.7167	12,589.4551	0.3764	583.6991
595	233.1283	12,449.1289	-0.6124	577.1930	651	238.0867	12,826.8838	0.4561	594.7072
596	238.4200	12,866.3311	0.5280	597.5562	652	237.9717	12,517.1475	0.4314	580.3466
597	235.8717	13,315.8564	-0.0212	617.8780	653	239.2367	12,833.1260	0.7040	594.9967
598	230.7950	12,905.5576	-1.1152	598.3549	654	235.2617	13,152.1484	-0.1526	609.7879
599	238.8433	13,053.3047	0.6192	605.2051	655	236.0533	12,355.2559	0.0180	572.8406
600	236.4450	12,322.2186	0.1024	571.3089	656	232.8717	13,056.2070	-0.6677	605.3396
601	236.5283	12,688.2344	0.1203	588.2789	657	232.5583	12,440.8701	-0.7352	576.8101
602	241.1150	13,133.5596	1.1088	608.9260	658	232.4133	12,778.7539	-0.7665	592.4757
603	235.7900	12,630.3662	-0.0388	585.5959	659	239.9233	13,023.8203	0.8520	603.8380
604	236.9083	13,426.1455	0.2022	622.4915	660	238.4600	12,713.1436	0.5366	589.4338
605	231.7733	12,892.1826	-0.9044	597.7346	661	232.7133	13,072.0049	-0.7018	606.0721
606	241.2483	12,672.5957	1.1375	587.5538	662	240.3600	12,889.3730	0.9461	597.6045
607	237.3700	12,612.8311	0.3017	584.7829	663	240.9833	12,554.9844	1.0804	582.1009
608	235.4567	12,905.6738	-0.1106	598.3603	664	237.3233	12,288.8838	0.2916	569.7693
609	238.2267	13,275.5283	0.4863	615.5083	665	236.9833	12,972.4795	0.2141	601.4577
610	237.1183	12,523.9355	0.2475	580.6613	666	235.7033	12,479.6045	-0.0575	578.6059
611	236.1950	13,049.7383	0.0485	605.0397	667	240.4383	13,481.5166	0.9629	625.0587
612	234.0950	12,556.2979	-0.4041	562.1618	668	237.7950	13,134.1084	0.3933	608.9514
613	239.2383	13,589.1230	0.7173	630.0478	669	236.6067	13,523.1641	0.1372	626.9897
614	240.8133	12,411.6709	1.0438	575.4563	670	243.4933	12,791.9102	1.6213	593.0857
615	241.6300	12,909.7324	1.2197	598.5484	671	238.8767	12,534.6758	0.6264	581.1593
616	237.8017	12,067.3467	0.3947	559.4920	672	231.6700	12,789.8613	-0.9267	592.9907

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร

ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
673	241.8817	12,522.5840	1.2740	580.5986	729	241.5300	12,949.2207	1.1982	600.3793
674	237.3400	12,639.6641	0.2952	586.0270	730	231.8450	12,384.4902	-0.8869	574.1961
675	239.1950	12,780.9600	0.6950	592.5780	731	237.9233	12,803.1992	0.4210	593.6091
676	238.6467	12,943.9883	0.5768	600.1367	732	242.0833	12,780.3340	1.3174	592.5490
677	238.0633	12,430.6807	0.4511	576.3376	733	236.9033	12,814.3809	0.2011	594.1276
678	233.5567	13,354.8086	-0.5201	619.1840	734	234.1850	12,763.3916	-0.3847	591.7635
679	235.2667	12,903.2813	-0.1516	598.2493	735	236.2783	13,039.6152	0.0664	604.5704
680	236.1833	12,884.1797	0.0460	597.3637	736	235.0133	13,103.5596	-0.2062	607.5351
681	238.6500	13,273.4336	0.5775	615.4111	737	235.4917	12,774.8330	-0.1031	592.2940
682	241.2550	12,440.3252	1.1389	576.7648	738	232.3283	12,924.3926	-0.7848	599.2282
683	231.2967	13,275.9053	-1.0071	615.5257	739	239.4117	12,856.7852	0.7417	596.1683
684	231.7500	12,987.0625	-0.9094	602.1336	740	231.8833	12,305.1953	-0.8807	570.5196
685	237.9117	13,316.9268	0.4184	617.4277	741	237.6300	12,547.0752	0.3577	581.7342
686	232.6533	13,290.0430	-0.7148	616.1812	742	233.8083	13,072.6709	-0.4658	606.1023
687	234.5667	12,564.6064	-0.3024	582.5470	743	239.3267	13,058.2842	0.7234	605.4359
688	234.6350	12,853.6133	-0.2877	595.9465	744	236.8850	13,623.8301	0.6282	631.6570
689	234.6350	12,863.4492	-0.2377	596.4026	745	237.9517	12,758.0215	0.4271	591.5145
690	232.7883	12,510.1826	-0.6857	580.0237	746	232.3967	12,975.5820	-0.7701	601.6015
691	236.4467	12,760.1738	0.1027	637.9784	747	241.0367	12,626.6836	1.0919	585.4251
692	235.9217	13,339.0586	-0.0104	618.4538	748	238.2333	12,461.1143	0.4978	577.7427
693	236.0483	13,532.1650	0.0169	627.4070	749	235.2667	12,854.5430	-0.1516	596.4533
694	236.3767	12,968.0010	0.0876	601.1573	750	236.7683	12,593.1660	0.1720	583.8711
695	237.8333	12,564.0654	0.4016	582.5219	751	234.3333	12,549.6387	-0.3527	581.9530
696	236.6333	12,950.2959	0.1429	600.4291	752	242.4850	12,678.2617	1.4040	587.8165
697	237.9517	12,763.2451	0.2978	591.7567	753	239.4867	12,929.7598	0.7579	599.4770
698	235.5500	13,772.4453	-0.0905	638.5474	754	235.2100	12,874.7402	-0.1638	596.9261
699	243.2633	12,476.6756	1.5728	578.5629	755	231.0250	12,924.8105	-1.0657	599.2475
700	234.5133	12,126.5879	-0.3139	562.2386	756	234.7200	13,394.2021	-0.2694	620.5468
701	233.4450	12,253.4609	-0.5441	568.1210	757	235.2867	12,504.9326	-0.1473	579.7803
702	226.5783	12,286.1162	-2.0239	569.6350	758	234.0017	12,819.3721	-0.4242	594.3590
703	232.6050	13,353.2227	-0.7252	619.1105	759	237.9150	12,301.4814	0.4192	570.3474
704	230.5267	12,559.4688	-1.1731	582.3088	760	234.7117	12,950.9316	-0.2712	600.4586
705	240.0933	12,759.9111	0.8886	591.6021	761	239.9050	12,847.3896	0.8480	595.6580
706	235.3017	13,272.5537	-0.1440	615.3703	762	232.1550	12,122.4805	-0.8221	562.0482
707	240.5583	12,992.7227	0.9888	602.5962	763	239.4767	12,973.9492	0.7557	601.5258
708	232.9467	12,553.1660	-0.6515	582.0166	764	236.4917	13,435.0898	0.1124	622.9062
709	235.8800	12,647.2842	-0.0194	586.9803	765	228.5783	12,742.2959	-1.5929	599.7854
710	237.0167	12,955.6660	0.2256	600.6781	766	236.4183	12,621.7412	0.0966	585.1960
711	238.9217	13,249.8936	0.6361	614.3197	767	233.0033	12,710.0498	-0.6393	589.2903
712	234.4450	13,070.4590	-0.3286	606.0004	768	228.6600	12,966.4551	-1.5753	601.1783
713	233.5917	13,140.2500	-0.5556	609.2362	769	232.6933	13,420.8271	-0.7061	622.2449
714	239.6000	13,420.1670	0.7823	622.2145	770	238.6017	12,629.1133	0.5671	585.5378
715	237.2800	12,996.5762	0.2823	602.5749	771	239.6650	13,184.0078	0.7963	611.2650
716	239.8750	12,336.6650	0.8415	571.9787	772	236.1650	12,510.5967	0.4730	590.0429
717	239.2550	13,100.3594	0.7079	607.3867	773	236.2300	11,922.7715	0.0560	552.7689
718	235.0750	13,228.0039	-0.1929	613.3048	774	232.8583	13,630.2734	-0.6706	631.9557
719	236.0917	12,764.4053	0.0262	591.8105	775	235.7483	12,811.9434	0.0178	591.0145
720	233.3300	13,265.8945	-0.5689	615.0616	776	235.2783	13,652.6533	-0.1491	632.9933
721	233.7017	13,134.9561	-0.4888	608.9907	777	235.9533	13,037.7139	-0.0036	604.4822
722	238.7000	12,444.5313	0.5883	576.9798	778	233.1483	12,450.5117	-0.6081	577.2571
723	236.1767	13,113.7383	-0.1710	608.0070	779	243.5567	13,240.0703	1.6349	613.8643
724	234.9700	12,407.0656	0.2115	577.2867	780	236.1117	12,981.1117	0.1483	610.2604
725	234.8467	12,763.9102	-0.2421	591.7875	781	239.1483	12,505.1563	0.6849	579.7906
726	240.3567	12,978.9648	0.9453	601.7583	782	235.4133	12,516.6230	-0.1200	580.3223
727	230.6550	12,804.8594	-1.1454	593.6861	783	237.3467	13,148.3672	0.2967	609.6125
728	238.1133	12,858.2646	0.4619	596.1622	784	236.7533	12,391.7256	0.1688	574.5315

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร
ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12.919.4717 กรณีที่ข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
785	233.5117	12,875.6543	-0.5298	596.9684	841	233.8783	13,092.5459	-0.4508	607.0244
786	239.1033	13,415.9160	0.6752	622.0172	842	232.0283	13,394.2559	-0.8494	621.0130
787	236.6600	12,502.1279	0.1487	579.6502	843	233.1033	12,546.5273	-0.6178	581.7088
788	243.0250	13,622.1865	1.5204	631.5808	844	230.1783	12,983.6943	-1.2481	601.9776
789	233.3200	13,042.7285	-0.5711	604.7147	845	234.7067	12,807.2725	-0.2723	593.7980
790	230.3817	12,891.9609	-1.2043	597.7245	846	237.2033	12,781.4277	0.2658	592.5997
791	237.1350	13,045.2471	0.2511	604.8315	847	242.8350	13,236.4170	1.4794	613.6949
792	241.8683	12,042.4131	1.2711	558.3359	848	234.8400	12,928.4551	-0.2435	599.4165
793	238.6533	12,840.7813	-0.4992	595.3516	849	236.7833	12,543.9600	0.1753	581.5897
794	234.7917	12,836.5342	-0.2539	595.1547	850	238.4217	12,799.2607	0.5283	593.4265
795	230.7650	12,399.8076	-1.1217	574.9062	851	233.0617	12,918.9248	-0.6268	598.9746
796	240.5183	12,721.8477	0.9802	589.8979	852	247.4250	12,799.5020	2.4686*	593.4377
797	239.6383	12,472.2100	0.7905	578.2631	853	232.7333	12,882.1523	-0.6975	597.2697
798	238.3983	13,075.9346	0.5233	606.2543	854	231.7667	12,770.5098	-0.9058	592.0935
799	235.5217	13,410.8760	-0.0966	621.7835	855	236.5800	12,781.0049	0.1315	592.5801
800	235.1967	13,200.1885	-0.1667	612.0152	856	235.3317	12,949.1348	-0.1376	600.3753
801	238.1017	13,032.2764	0.4594	604.2301	857	231.8450	13,054.8877	-0.8889	605.2784
802	235.1250	13,381.6436	-0.1821	620.4282	858	239.3833	12,242.4238	0.7356	567.6093
803	230.4300	13,510.0654	-1.1939	626.3824	859	232.6117	12,877.2080	-0.7237	597.0405
804	238.8450	13,294.8572	0.6196	616.3951	860	237.0633	13,399.0547	0.2356	621.2354
805	231.8033	13,231.8340	-0.8979	613.4824	861	238.1617	12,918.8057	0.4723	598.9691
806	236.7500	12,941.8936	0.1681	600.0396	862	237.9033	12,490.1279	0.4166	579.0939
807	233.2517	12,352.5742	-0.5858	572.7163	863	238.7950	12,623.3301	0.8243	585.2697
808	232.7950	13,128.4424	-0.6842	608.6887	864	238.3567	13,176.6807	0.5143	610.9253
809	233.5467	12,390.7324	-0.5222	574.4855	865	235.5367	12,397.7813	-0.0934	574.8123
810	242.0700	14,108.1689	1.3146	654.1129	866	235.6583	12,528.6113	-0.0672	580.8781
811	235.6267	12,778.2246	-0.0740	592.4512	867	233.9300	12,775.5674	-0.4396	592.3280
812	240.7417	12,634.5879	1.0283	585.7916	868	239.2517	13,363.6465	0.7072	619.5938
813	234.1850	12,585.7422	-0.3847	583.5269	869	233.7300	12,952.8223	-0.4827	600.5463
814	234.3267	12,775.4658	-0.3541	592.3233	870	234.6017	12,456.4658	-0.2949	577.5331
815	237.0783	13,286.8818	0.2388	616.0347	871	242.2017	13,355.1709	1.3429	619.2008
816	237.8050	13,465.4648	0.3954	624.3145	872	238.5917	12,629.8320	0.5650	585.5479
817	234.2767	13,130.0264	-0.3649	608.7622	873	238.2867	12,773.9072	0.4992	592.2510
818	236.9267	13,470.1445	0.2062	624.5315	874	231.4233	13,012.1611	-0.9798	603.2975
819	231.4117	12,362.5313	-0.9823	573.1779	875	234.4983	12,916.0186	-0.3171	598.8399
820	235.3783	13,187.3662	-0.1275	611.4207	876	231.2967	12,871.2578	-1.0071	596.7646
821	235.8817	13,428.0127	-0.0190	622.5781	877	232.0217	12,447.1777	-0.8509	577.1025
822	230.7583	13,122.4961	-1.1231	608.4130	878	232.0400	13,304.0117	-0.8469	616.8289
823	234.4417	13,310.2354	-0.3294	617.1174	879	236.7000	13,143.3223	0.1573	609.3786
824	235.5333	12,428.9570	-0.0941	576.2577	880	237.2833	13,031.5859	0.2830	604.1981
825	236.2550	13,156.7295	0.0614	610.0003	881	239.5867	13,301.1445	0.7794	616.6959
826	236.7133	13,358.3955	0.1602	619.3503	882	235.7717	12,994.2314	-0.0427	602.4662
827	234.8900	12,179.3135	-0.2327	564.6832	883	239.0517	13,310.1240	0.6641	617.1123
828	238.9617	12,931.2402	0.6447	599.5456	884	231.0550	12,915.2061	-1.0592	598.8022
829	233.8050	12,617.5713	-0.4666	585.0027	885	231.4117	12,973.8037	-0.9623	601.5191
830	230.1817	12,904.1094	-1.2582	598.2877	886	234.0367	12,677.7080	-0.4168	587.7908
831	238.6817	14,118.4473	0.5844	654.6358	887	230.8350	13,185.0645	-1.1066	611.3140
832	238.8683	13,319.8424	-0.4529	617.5675	888	236.1150	12,589.7832	0.0312	583.7143
833	239.3317	13,176.2402	0.7244	610.9048	889	236.9000	13,676.1035	0.2004	634.0806
834	234.3833	13,231.8857	-0.3419	613.4848	890	239.2150	12,342.7412	0.6993	572.2604
835	232.0817	13,003.4453	-0.8379	602.8934	891	235.0367	12,946.8369	-0.2011	600.2688
836	232.4417	13,515.9785	-0.7604	626.6565	892	240.0667	13,059.0674	0.8828	605.4722
837	240.8583	12,712.2500	1.0535	589.3923	893	238.9867	12,864.1084	0.6501	596.4330
838	238.7967	12,384.1660	0.6092	574.1810	894	235.4583	13,021.2285	-0.1103	603.7179
839	231.1633	13,520.6074	-1.0359	626.8711	895	239.5350	13,159.9365	0.7683	610.1489
840	233.4150	13,416.9795	-0.5506	622.0665	896	240.3883	12,481.0908	0.9522	578.6749

ตาราง 28 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบง่าย กับค่าคะแนนเฉลี่ยของประชากร
ประชากรซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีการแจกแจง

แบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

No.	Mean	Variance	t-value	Chi-square	No.	Mean	Variance	t-value	Chi-square
897	234.6133	13,567.6836	-0.2924	629.0538	953	241.0800	12,564.2402	1.1012	582.5300
898	235.0100	12,952.5078	-0.2069	600.5317	954	236.0867	12,796.2666	0.0251	593.2877
899	236.1167	12,515.1387	0.0316	580.2535	955	232.2817	12,588.3613	-0.7948	583.6484
900	238.5067	12,315.2148	0.5467	570.9842	956	239.5933	12,695.7012	0.7808	588.6251
901	236.9500	12,459.9775	0.2112	577.6960	957	238.1217	12,587.8184	0.4637	583.6232
902	234.8517	12,925.8379	-0.2410	599.2952	958	231.2217	13,115.3594	-1.0233	608.0822
903	239.5833	12,943.9932	0.7787	600.1369	959	234.8517	12,802.0117	-0.2410	593.5541
904	241.5967	12,801.4463	1.2126	593.5279	960	237.3267	13,175.0518	0.2924	610.8497
905	230.4850	13,488.8730	-1.1820	625.8998	961	237.0433	13,398.8428	0.2313	621.2256
906	232.9783	12,345.9824	-0.6458	572.4107	962	231.7183	13,572.3877	-0.9162	629.2719
907	233.2967	12,405.6348	-0.5761	575.1764	963	234.1617	13,157.4033	-0.3897	610.0315
908	240.3900	12,509.6836	0.9525	580.0005	964	236.5533	13,125.9102	0.1257	608.5713
909	239.6283	13,064.4492	0.7884	605.7218	965	233.8850	13,153.6230	-0.4493	609.8562
910	237.0683	13,559.4180	0.2367	628.6705	966	233.4317	12,934.2939	-0.5470	599.6872
911	231.2750	13,060.6484	-1.0118	605.5455	967	235.8333	13,168.4063	-0.0295	610.5416
912	239.6050	13,227.0557	0.7834	613.2609	968	228.6500	12,636.8789	-1.5775	585.8978
913	236.0117	13,434.1035	0.0090	622.8605	969	238.2117	12,656.7869	0.4831	586.8213
914	238.5167	13,269.6963	0.5488	615.2379	970	235.3150	12,777.2598	-0.1412	592.4065
915	238.3800	12,785.0010	0.5194	593.2290	971	232.8300	12,604.4648	-0.6767	584.3950
916	232.3000	13,006.5947	-0.7909	603.0394	972	233.7200	13,962.4727	-0.4849	647.3578
917	238.2483	13,063.2969	0.4910	605.6683	973	234.4717	13,347.5771	-0.3229	618.8487
918	242.5833	13,440.6406	1.4252	623.1635	974	231.9833	12,769.8096	-0.8591	592.0610
919	231.3883	12,473.3477	-0.9874	578.3159	975	239.2467	12,412.9404	0.7061	575.5151
920	245.8567	12,573.9092	2.1306*	582.9783	976	236.0450	12,877.3398	0.0162	597.0466
921	230.8550	12,674.4795	-1.1023	587.6412	977	239.4933	13,293.3389	0.7593	616.3340
922	236.8000	12,537.1553	0.1789	581.2742	978	235.8267	12,885.8193	-0.0309	597.4397
923	234.9383	13,067.7461	-0.2223	605.8746	979	239.4083	13,309.1787	0.7410	617.0684
924	234.1150	13,905.8887	-0.3998	616.8927	980	243.7017	12,711.0557	1.6662	589.3370
925	235.4283	12,905.5693	-0.1167	598.3554	981	230.9183	13,342.6680	-1.0886	618.6211
926	238.8850	12,625.2803	0.6239	585.3601	982	240.1817	12,883.6533	0.9076	597.3393
927	238.2550	11,981.0303	0.4924	555.4900	983	235.5383	13,237.7295	-0.0930	613.7557
928	239.6750	13,141.2168	0.7984	609.2810	984	234.5467	12,521.4102	-0.3067	580.5442
929	237.9783	13,336.5137	0.4328	618.3358	985	240.1567	12,971.5078	0.9022	601.4126
930	234.0500	12,706.9287	-0.4138	589.1456	986	233.7833	13,025.5625	-0.4712	603.9188
931	234.9017	12,864.3408	-0.2302	596.4439	987	238.5317	13,215.5898	0.6520	612.7293
932	236.3433	13,015.4209	0.0805	603.4486	988	239.7883	13,129.6650	0.8229	608.7454
933	235.6217	13,094.8350	-0.0751	607.1306	989	244.0833	13,328.0928	1.7484	617.9454
934	231.8950	12,620.8545	-0.8566	585.1549	990	241.9150	12,767.5254	1.2812	591.8551
935	234.5583	13,100.0078	-0.3042	607.3704	991	237.0117	13,459.5957	0.2245	624.0424
936	236.2017	12,725.7021	0.0499	590.0160	992	234.4417	13,094.2148	-0.3294	607.1018
937	235.4183	13,116.5029	-0.1189	608.1352	993	235.4900	12,888.2803	-0.1034	597.5538
938	230.9317	12,057.4678	-1.0858	559.0939	994	235.4667	13,388.3164	-0.1085	620.7376
939	239.0617	13,169.0010	0.6663	610.5692	995	237.8183	13,239.9131	0.3983	613.8570
940	234.4350	12,676.7588	-0.3308	587.7468	996	229.8700	12,363.1885	-1.3146	573.2084
941	237.2983	13,917.6309	0.2863	645.2788	997	233.1150	12,344.8922	-0.6153	572.9648
942	231.0533	12,721.3262	-1.0596	589.8132	998	231.7883	12,914.1191	-0.9012	598.7518
943	236.9433	12,275.6025	0.2098	568.1476	999	228.6283	12,670.1152	-1.5822	587.4388
944	231.9183	12,735.4121	-0.8731	590.4662	1000	232.4550	13,013.0078	-0.7575	603.3367
945	233.9133	13,118.5029	-0.4432	608.2279					
946	237.0267	13,195.9727	0.2277	611.8197					
947	233.3833	12,989.1680	-0.5574	602.2314					
948	234.7933	13,209.9268	-0.2536	612.4867					
949	236.5150	12,979.8611	0.1174	601.7906					
950	237.4167	13,038.4902	0.3118	604.5182					
951	239.8600	11,925.0352	0.8383	552.8938					
952	241.5017	12,961.7344	1.1821	600.9595					
TOTAL	236.0827	12,938.2236	0.0243	599.7767					

ตาราง 28. สมรรถนะของพืชและดินภายใต้ระบบการปลูกแบบผสมผสานที่สุ่มโดยวิธีการสุ่มแบบหลายขั้นตอน กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,949.4717 กรณีนี้ข้อสมมติลักษณะประชากรแจกแจงค่า

ความยาวของข้อมูลแบบปกติ จำนวนตัวอย่างสุ่มเท่ากับ 800 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
1	235.9817	12,949.5527	116.1206	0.0025	600.3947	57	235.9533	13,011.9131	115.9409	-0.0036	603.2860
2	235.9267	13,025.8057	115.9985	-0.0093	603.9301	58	236.3683	12,975.9209	116.1187	0.0857	601.6172
3	235.9550	12,905.9502	116.2517	-0.0032	598.3731	59	235.6867	12,909.6064	116.0249	-0.0611	598.5426
4	235.9000	12,892.4902	116.1751	-0.0151	597.7490	60	235.9650	12,891.6768	116.0376	-0.0011	597.7113
5	235.3617	12,984.7656	116.0720	-0.1265	602.0273	61	236.4583	12,891.1729	116.2359	0.1054	597.6879
6	236.1367	12,859.5020	116.0989	0.0360	596.2196	62	236.1050	12,913.0400	116.0371	0.0291	598.7018
7	235.8317	12,856.4795	116.2344	-0.0299	596.0794	63	235.9200	12,973.1592	116.1251	-0.0108	601.4892
8	236.1317	13,012.0283	116.2042	0.0347	603.2913	64	235.7233	12,875.6299	116.1979	-0.0532	596.9673
9	235.7917	12,841.2891	116.3096	-0.0386	595.3751	65	235.6267	12,912.1768	116.2256	-0.0740	598.6618
10	236.1000	12,898.3154	116.2311	0.0261	595.2373	66	236.5217	12,912.5283	116.1367	0.1189	598.6781
11	236.0433	12,872.4854	116.1997	0.0158	596.8215	67	236.1983	12,961.0576	116.0830	0.0491	600.9281
12	236.1100	12,914.5039	116.0886	0.0302	598.7697	68	236.1000	12,832.2754	116.1968	0.0281	594.9572
13	235.7600	12,937.7900	116.1491	-0.0452	599.8493	69	235.7783	12,965.3135	116.1255	-0.0412	601.1254
14	236.4900	12,877.5742	116.1004	0.1122	597.0575	70	236.0633	12,926.1299	116.1344	0.0201	599.3087
15	235.7983	12,895.7041	116.2021	-0.0370	597.8980	71	235.8367	12,926.6309	116.1005	-0.0287	599.3319
16	236.1767	12,944.8201	116.1504	0.0445	600.1757	72	236.1467	12,975.9092	116.1320	0.0380	601.6167
17	236.0467	13,016.6621	116.1164	0.0165	603.5061	73	236.1400	12,884.3174	116.1772	0.0367	597.3701
18	236.0567	12,947.7822	116.0998	0.0208	600.8126	74	236.0367	12,862.0527	116.2259	0.0144	596.3378
19	235.6517	12,867.5527	116.0070	-0.0255	601.2292	75	235.7767	12,939.0879	116.2058	-0.0416	599.9095
20	236.0750	12,813.8428	116.0966	0.0227	594.1026	76	235.8867	12,921.7129	116.1914	-0.0180	599.1039
21	235.8050	12,976.5068	116.0715	-0.0355	601.6444	77	236.2417	12,913.2529	116.2007	0.0536	598.7117
22	236.0450	12,896.1455	115.9971	0.0162	597.9185	78	235.6883	12,945.4873	116.1958	-0.0606	600.2062
23	235.9400	12,871.7275	116.2111	-0.0065	596.7864	79	236.1200	12,875.7949	116.1483	0.0324	596.9750
24	236.0400	12,961.2891	116.1273	0.0151	600.9388	80	235.9267	12,920.6326	116.1941	-0.0093	599.0538
25	235.8333	12,904.8750	116.1502	-0.0295	598.3232	81	236.1650	12,849.5137	116.0981	0.0421	595.7565
26	236.3183	12,979.3838	116.3054	0.0749	601.7778	82	235.7817	12,852.6250	116.2467	-0.0407	595.9007
27	236.1850	12,895.6406	116.1367	0.0464	597.8024	83	236.0017	12,963.6553	116.1330	0.0068	601.0485
28	235.4250	12,871.2764	116.2704	-0.1177	596.7655	84	235.8200	12,921.2949	116.2434	-0.0323	599.0845
29	236.8550	12,955.0645	116.1162	-0.0247	600.6502	85	236.1817	12,988.3887	116.0085	0.0455	602.6589
30	236.1117	12,926.7012	116.0407	0.0305	599.3352	86	236.5467	12,910.2949	116.3475	0.1243	598.5745
31	236.1967	12,949.4951	116.0995	0.0486	600.3920	87	235.9083	12,986.3457	116.1037	-0.0133	602.1006
32	235.9450	12,956.9854	116.1634	-0.0054	600.7393	88	235.9483	12,917.3037	116.2684	-0.0047	598.8995
33	235.8717	12,957.6230	116.1875	-0.0642	600.7689	89	236.0617	12,900.6748	116.2345	0.0198	598.1285
34	235.9350	12,940.4502	116.1143	-0.0075	599.9727	90	235.9133	12,920.0898	116.0979	-0.0122	599.0287
35	235.8633	12,976.1543	115.9677	-0.0219	601.7208	91	235.5750	12,868.5195	116.2753	-0.0852	597.5649
36	236.2400	12,790.0708	116.2240	0.0585	593.0004	92	236.1433	12,964.2227	116.0601	0.0373	601.0748
37	235.5700	12,874.2207	116.2973	-0.0864	596.9020	93	235.6617	12,870.7549	116.1783	-0.0666	596.7413
38	235.6133	12,895.4512	116.0825	-0.0769	597.8863	94	235.7867	13,015.3945	116.0856	-0.0394	603.4474
39	235.6050	12,922.6328	116.1694	-0.0786	599.1929	95	236.1067	12,723.3350	116.1059	0.0297	589.9063
40	235.8617	12,927.8359	116.1988	-0.0233	599.3878	96	235.7600	12,940.1309	116.1755	-0.0452	599.9578
41	235.9833	12,876.5205	116.0896	0.0050	601.6450	97	235.9833	12,960.7617	116.2047	0.0029	600.9144
42	235.7150	12,937.1211	116.1084	-0.0549	599.8183	98	236.0317	12,887.2373	116.2459	0.0133	597.5055
43	235.9600	12,958.0430	116.1748	-0.0022	600.7883	99	236.3217	12,985.4063	116.0644	0.0756	602.0570
44	236.3400	12,898.4736	116.1967	0.0798	598.0264	100	235.8550	12,927.6006	115.9943	-0.0248	599.3769
45	236.1617	12,981.1631	116.0637	0.0412	601.8603	101	236.1367	12,930.1670	116.2160	0.0359	599.4959
46	235.8783	12,989.8545	116.0521	-0.0197	602.2632	102	236.2217	12,965.3311	115.9543	0.0541	601.1262
47	235.9533	12,953.2500	116.1501	-0.0036	600.5661	103	235.9783	13,021.1895	116.2077	0.0018	603.7161
48	235.7233	12,862.1104	116.0971	-0.0532	597.2678	104	236.1483	12,971.1729	116.0698	0.0384	601.3971
49	236.0900	12,864.1436	116.1852	0.0259	596.4346	105	235.7017	12,837.0615	116.1892	-0.0580	595.1791
50	235.6600	12,857.6709	116.1443	-0.0670	595.1347	106	236.2767	12,883.6895	116.2265	0.0662	597.3410
51	235.9183	12,835.6729	116.1810	-0.0112	595.1147	107	236.2783	12,972.8555	116.1034	0.0663	601.4751
52	235.8217	12,917.9492	116.1014	-0.0320	598.9294	108	236.1750	12,933.0244	115.9535	0.0442	599.6284
53	236.1183	12,996.9277	116.0698	0.0319	602.5912	109	235.8550	12,922.4561	116.2860	-0.0032	599.1384
54	235.9017	12,949.6230	116.0994	-0.0147	600.3979	110	235.7917	12,898.5674	116.0800	-0.0385	598.0308
55	236.2667	12,950.2900	116.2375	0.0639	600.4289	111	235.6033	12,926.4346	116.1227	-0.0790	599.3228
56	236.0833	12,968.0420	116.0736	0.0244	601.2519	112	235.5567	12,845.2676	116.2630	-0.0893	595.5596

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 235.9700 และความแปรปรวนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีลักษณะการแจกแจงค่า

ความยากของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
113	236.2417	12,975.0684	116.0787	0.0584	601.5777	169	236.0950	12,911.7373	116.1182	0.0269	598.6414
114	236.2150	13,009.9551	116.2176	0.0526	603.1952	170	236.0983	12,944.2373	116.0880	0.0276	600.1482
115	235.7550	12,826.9033	116.0689	-0.0465	594.7081	171	236.0917	12,898.7988	116.0270	0.0262	598.0415
116	235.9683	12,802.7764	116.1754	-0.0004	593.5895	172	235.7517	12,951.1221	116.1939	-0.0470	600.4674
117	235.4617	13,028.7520	116.0216	-0.1091	604.0667	173	236.3217	13,028.0313	116.1301	0.0755	604.0333
118	236.1317	12,918.3301	116.1786	0.0349	598.3443	174	236.0017	12,940.1035	116.1414	0.0068	599.9566
119	235.6517	12,918.9032	116.2861	-0.0686	598.9739	175	236.0533	13,023.1709	116.0390	0.0179	603.8079
120	236.2567	12,906.4082	115.9579	0.0618	598.3943	176	235.9367	12,948.0469	116.1779	-0.0072	600.3249
121	235.9267	12,925.1709	116.1458	-0.0093	599.2642	177	235.9083	12,995.7803	116.0846	-0.0132	602.5380
122	236.0433	12,910.8789	116.2621	0.0158	598.6016	178	235.7033	12,988.0684	116.0493	-0.0573	602.1803
123	236.0200	12,901.9258	116.2155	0.0108	598.1865	179	236.0933	12,917.9525	116.1239	0.0266	598.9017
124	236.1233	12,856.8359	116.1847	0.0331	596.0959	180	235.4900	12,889.9053	116.1591	-0.1057	597.6232
125	236.1950	12,954.2178	116.1912	0.0484	600.6110	181	235.9033	12,893.7314	116.0534	-0.0144	597.8066
126	236.1283	12,930.1641	116.0918	0.0341	599.4957	182	236.0150	12,827.6973	116.1916	0.0097	594.7450
127	236.1833	12,927.5449	116.1762	0.0460	599.3743	183	235.6500	12,854.6318	116.1097	-0.0691	595.9338
128	235.7800	12,859.4521	116.1718	-0.0389	596.2172	184	236.1733	12,840.0254	116.3140	0.0440	595.3165
129	235.7883	12,875.0361	116.1776	-0.0371	596.9398	185	236.1750	12,940.3613	116.1258	0.0441	599.9685
130	235.5950	12,946.7666	116.2470	-0.0807	600.2655	186	235.6050	12,858.6133	116.0780	-0.0788	596.1784
131	236.3033	12,927.0557	116.0997	0.0718	599.3516	187	235.8483	12,887.2578	116.1761	-0.0263	597.5064
132	235.9300	12,773.0586	116.1656	-0.0087	592.2117	188	235.8150	12,998.9854	116.0147	-0.0333	602.6866
133	236.1533	12,919.2373	116.0564	0.0395	598.9891	189	235.9917	12,952.6436	116.1689	0.0047	600.5380
134	236.0367	13,008.0186	116.1955	0.0143	603.1054	190	235.8200	12,971.1504	116.0605	-0.0323	601.3960
135	236.0200	12,935.3711	116.1856	0.0108	599.7372	191	236.1783	12,906.4463	116.1182	0.0449	598.3961
136	236.1800	12,908.8287	116.2453	0.0453	598.5112	192	236.6550	12,903.7764	116.1439	0.1477	598.2723
137	235.9800	12,969.2961	116.1822	0.0022	601.3096	193	235.9117	12,912.9160	116.2340	-0.0126	598.6961
138	235.2917	12,898.7402	116.1478	-0.1463	598.0388	194	235.7500	12,962.8584	116.1994	-0.0473	601.0116
139	235.8017	12,829.9658	116.1485	-0.0363	599.4866	195	236.1817	12,939.6523	116.1069	0.0456	599.9357
140	235.6617	12,943.2852	116.2083	-0.0664	600.1041	196	235.6633	12,919.5674	116.2577	-0.0661	599.0044
141	235.8083	12,850.8457	116.1106	-0.0349	595.8182	197	236.1433	12,929.3301	116.2503	0.0373	599.4571
142	235.8233	12,829.7939	116.4480	-0.0317	594.8422	198	235.7817	12,826.0449	116.2239	-0.0407	594.6684
143	235.9017	12,873.7285	116.1379	-0.0148	596.8792	199	235.6600	12,937.1279	116.2154	-0.0668	599.8186
144	235.6917	12,943.9248	116.1837	-0.0599	600.1337	200	235.8467	12,884.7061	116.2559	-0.0266	597.9881
145	236.0183	12,906.7900	116.2768	0.0104	598.4120	201	236.3183	12,905.7783	116.1811	0.0751	598.3651
146	236.1883	12,920.8535	116.1433	0.0470	599.0641	202	235.6833	12,997.1816	116.0873	-0.0616	602.8030
147	235.8150	12,975.7959	116.0276	-0.0333	601.6114	203	235.8783	12,902.6650	116.2570	-0.0198	598.2208
148	236.4200	13,016.7854	116.0746	0.0966	603.5095	204	236.1583	12,959.5684	116.1452	0.0405	600.8590
149	235.3467	12,990.5352	116.1322	-0.1340	602.2948	205	235.8317	12,911.5625	116.1944	-0.0298	598.6333
150	235.7300	12,919.6055	116.1391	-0.0517	599.0062	206	235.7750	12,926.4756	116.1861	-0.0420	599.3247
151	235.8217	12,916.0527	116.1150	-0.0320	598.8415	207	236.1500	12,941.3574	116.1627	0.0388	600.0147
152	235.8200	12,897.2266	116.3267	-0.0108	597.9686	208	236.1617	12,948.4053	116.1859	0.0413	600.3415
153	236.0900	12,932.1836	116.0578	0.0258	599.5894	209	235.9417	12,934.0928	116.1495	-0.0061	599.6779
154	236.2700	12,845.4258	116.2021	0.0648	595.5669	210	236.2117	13,009.8262	116.1046	0.0519	603.1892
155	236.0467	12,909.8213	116.1802	0.0165	598.5526	211	235.9950	12,939.0059	116.1050	0.0054	599.9057
156	236.0683	12,842.9561	116.1803	0.0213	595.4524	212	236.0167	12,963.6133	116.0130	0.0100	601.0466
157	235.8683	12,888.4551	116.1723	-0.0219	597.5619	213	235.8367	12,969.4600	116.1142	-0.0287	601.3177
158	236.1684	13,047.9619	116.0672	0.0425	604.9573	214	236.1967	12,827.9404	116.2546	0.0490	594.7562
159	235.9683	12,934.9219	116.1849	-0.0004	599.7163	215	236.0817	12,901.8320	116.0563	0.0241	598.1822
160	235.6300	12,956.4004	116.0677	-0.0732	600.7122	216	235.5700	12,949.6230	116.1669	-0.0861	600.3979
161	235.8500	12,887.2275	116.1072	-0.0259	597.5050	217	235.9700	12,842.5107	116.3184	0.0000	595.4318
162	236.2450	12,976.3691	116.1049	0.0591	601.6380	218	235.9283	12,955.1826	116.0594	-0.0090	600.8557
163	236.0983	12,813.6025	116.1819	0.0278	594.0915	219	236.3800	12,875.2588	116.1557	0.0885	596.9501
164	235.8900	12,940.9795	116.1728	-0.0172	599.9972	220	236.0983	12,937.8027	116.1499	0.0266	599.8499
165	236.5183	12,951.2637	116.0746	0.1180	600.4740	221	235.9000	12,966.8506	116.1103	-0.0151	601.1967
166	235.6683	12,958.7715	116.0829	-0.0649	600.8221	222	236.0567	12,829.3174	116.2966	0.0187	594.8201
167	236.0783	12,895.4756	116.1970	0.0234	597.8874	223	235.8500	12,813.1094	116.0444	-0.0043	594.0686
168	236.0700	12,958.2119	116.1703	0.0215	600.7962	224	236.0300	12,925.8564	116.1625	0.0129	599.2960

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 236.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีนี้ข้อสอบมีลักษณะการแจกแจงค่า

ความยาวของข้อสอบแบบปกติ จำนวนตัวอย่างที่ตอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
225	236.2533	12,857.8203	116.2558	0.0612	596.1416	281	236.0750	12,945.9766	116.1089	0.0226	600.2289
226	235.8067	12,945.6250	116.0402	-0.0352	600.2126	282	236.2100	12,946.7168	116.1285	0.0517	600.2632
227	236.0617	13,000.6641	116.0800	0.0197	602.7644	283	236.0950	12,867.8438	116.2510	0.0270	596.6063
228	235.7600	12,868.8242	116.2042	-0.0453	596.6518	284	235.7700	12,880.7813	116.1994	-0.0432	597.2062
229	236.3100	12,944.2959	116.1779	0.0732	600.1510	285	236.0067	12,940.5791	116.1133	0.0079	599.9786
230	236.6833	12,948.1172	116.1719	0.1536	600.3281	286	236.1467	12,923.5684	116.2192	0.0381	599.1899
231	236.3750	12,869.4453	116.2521	0.0874	596.6806	287	236.0667	12,897.6172	116.2396	0.0208	597.9867
232	235.4700	12,949.4326	116.1237	-0.1076	600.3891	288	236.3050	12,928.8926	116.1419	0.0722	599.4368
233	236.0800	12,948.2705	116.0673	0.0237	600.3352	289	236.1567	12,981.2197	116.1033	0.0401	599.0629
234	235.7750	12,862.8750	116.2594	-0.0421	596.3759	290	235.8350	12,872.9727	116.1773	-0.0291	596.8441
235	236.1367	12,920.5156	116.1859	0.0359	599.0484	291	236.1883	12,890.9023	115.9511	0.0471	597.6754
236	235.9633	12,798.5986	116.0378	-0.0014	593.3958	292	236.1000	12,919.0605	116.1339	0.0280	598.9809
237	236.0567	12,991.7734	116.0899	0.0186	602.3522	293	235.9433	12,931.7432	116.1941	-0.0057	599.5690
238	235.9217	12,894.1309	116.2068	-0.0104	597.8251	294	236.1850	12,971.8926	116.2118	0.0462	601.4304
239	236.3400	12,841.8281	116.1994	0.0800	595.4001	295	236.3817	12,920.9678	116.0090	0.0887	599.0694
240	236.0833	12,988.5869	116.1059	0.0244	602.2045	296	236.2117	12,971.0410	116.0514	0.0520	601.3910
241	235.5750	12,873.8242	116.1629	-0.0853	596.8836	297	236.3000	12,872.2686	116.1529	0.0712	596.8115
242	236.3483	12,983.4766	116.0338	0.0813	601.9675	298	235.8967	12,848.4785	116.1836	-0.0158	595.7085
243	235.6900	12,942.5791	116.0816	-0.0603	600.0714	299	235.9050	13,001.1094	116.0151	-0.0140	602.7851
244	235.5717	12,939.5049	116.1217	-0.0858	599.9288	300	235.8867	12,882.5645	116.1425	-0.0180	597.2888
245	235.7750	12,893.3193	116.2126	-0.0421	597.7875	301	236.1850	12,935.6328	116.1827	0.0463	599.7493
246	235.5350	12,971.4756	116.1893	-0.0936	601.4111	302	235.7917	12,904.4805	116.1670	-0.0385	598.3049
247	236.3183	12,955.5850	116.1754	0.0750	600.6744	303	235.5267	12,923.0371	116.1822	-0.0955	599.1653
248	235.8033	12,970.1758	116.1026	-0.0358	601.3509	304	236.0650	12,872.1709	116.1774	0.0205	596.8069
249	236.1200	12,853.2207	116.1831	0.0324	595.9283	305	235.7883	12,993.4854	116.1222	-0.0390	602.4316
250	236.4317	12,835.4697	116.1716	0.0998	595.1053	306	236.2533	12,879.0801	116.0891	0.0612	597.1273
251	235.8833	12,880.8379	116.1146	-0.0187	597.2088	307	235.9250	12,921.5342	116.2081	-0.0097	599.0953
252	235.8533	12,971.8662	116.0696	-0.0251	601.4292	308	236.1450	12,944.8252	116.1991	0.0377	600.1755
253	235.9817	12,906.5479	116.2603	0.0025	598.4008	309	236.0450	12,858.5625	116.2696	0.0162	596.1760
254	235.5133	12,867.0439	116.1281	-0.0986	596.5692	310	236.2267	12,856.2852	116.2292	0.0554	596.0704
255	236.2583	12,818.6211	116.1917	0.0624	594.3242	311	235.7783	12,842.7227	116.1982	-0.0414	595.4416
256	235.9783	12,924.2939	116.1338	0.0018	599.2236	312	235.5900	12,994.7588	116.1696	-0.0817	602.4906
257	236.2417	12,944.0078	116.0702	0.0585	600.1376	313	235.9400	12,935.2637	116.1044	-0.0065	599.7322
258	236.3133	12,967.8340	115.9878	0.0738	601.2423	314	236.0567	12,850.4160	116.2215	0.0187	595.7983
259	236.0417	12,930.6631	116.1565	0.0154	599.5189	315	236.3350	12,953.1924	116.1808	0.0786	600.5634
260	236.0267	12,896.7578	116.1933	0.0122	597.9469	316	235.9767	12,867.9551	116.2464	0.0014	596.6115
261	235.6983	12,946.7539	116.0986	-0.0585	600.2649	317	235.9017	12,998.2900	116.0716	-0.0147	602.6543
262	236.0500	12,919.2109	116.1660	0.0172	598.9879	318	236.0400	12,890.6230	116.0604	0.0151	597.6625
263	236.1800	12,884.4932	116.0480	0.0453	597.3783	319	235.9717	12,923.1572	116.2053	0.0004	599.1709
264	235.8900	12,855.7666	116.2694	-0.0173	596.0464	320	235.9950	12,919.2588	116.2943	0.0054	598.9901
265	236.0033	12,861.4326	116.1882	0.0072	596.3091	321	235.9150	12,939.7852	116.2618	-0.0118	599.9418
266	235.7500	12,899.3496	116.1487	-0.0474	598.0671	322	236.1517	12,943.9902	116.0529	0.0391	600.1368
267	236.1650	12,976.7539	116.1294	0.0419	601.6558	323	235.5350	12,929.0840	116.0660	-0.0937	599.4457
268	235.8583	12,865.6523	116.3983	-0.0241	596.5047	324	235.8517	12,867.9619	116.2301	-0.0256	596.6118
269	235.8433	12,879.5850	116.2504	-0.0273	597.1507	325	235.7650	12,883.3818	116.2972	-0.0442	597.3267
270	236.1900	12,913.9521	116.1724	0.0474	598.7441	326	236.3283	12,962.3193	116.1018	0.0771	600.9866
271	235.8950	12,956.0781	116.0826	-0.0161	600.6972	327	235.7617	12,964.8662	116.2253	-0.0448	601.1047
272	235.9717	12,906.5566	116.2545	0.0004	598.4012	328	235.9350	12,988.7842	116.1087	-0.0075	602.2136
273	236.3633	12,871.4727	116.1901	0.0849	596.7746	329	235.7750	13,019.6123	116.0953	-0.0419	603.6429
274	236.0133	12,888.9590	116.3456	0.0093	597.5853	330	235.5883	12,976.1563	116.1094	-0.0821	601.6281
275	235.7950	12,837.9697	116.2519	-0.0378	595.2212	331	235.8417	12,932.4678	116.1556	-0.0276	599.6026
276	235.7433	12,900.5313	116.1677	-0.0489	598.1218	332	236.4283	12,907.7588	116.1231	0.0988	598.4569
277	235.6450	12,904.3672	116.2382	-0.0701	598.2997	333	235.8300	12,839.4834	116.2236	-0.0303	595.2914
278	235.9917	12,928.6035	116.2037	0.0047	599.4234	334	235.9017	12,863.6680	116.1563	-0.0148	596.4127
279	235.8833	12,877.5234	116.2588	-0.0187	597.0551	335	236.4500	12,851.1064	116.2138	0.1037	595.8303
280	235.8117	13,001.3955	116.0702	-0.0340	602.7983	336	235.9217	12,828.4834	116.2262	-0.0105	594.7814

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่ผู้สมัครสอบวิชาภาษาอังกฤษ กับ ค่าคะแนนเฉลี่ยของปี ๖๖-๖๘
ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีนี้ผู้สอบมีลักษณะการแจกแจงค่า

ความยาวของคำศัพท์ตามปกติ						จำนวนตัวอักษรผู้สอบเท่ากับ 600 คน					
No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
337	235.9883	12,938.4541	116.1750	0.0039	599.8801	393	236.1100	12,886.0586	116.1939	0.0302	597.4508
338	236.1767	12,933.9531	116.2084	0.0445	599.6714	394	236.0183	12,869.8760	116.2029	0.0104	596.7005
339	235.8483	12,989.3555	116.1281	-0.0262	602.2401	395	235.9550	12,832.2295	116.1207	-0.0032	594.9551
340	235.9950	12,863.9775	116.1821	0.0054	596.4271	396	236.1383	12,915.8672	116.2397	0.0363	598.8329
341	236.2483	12,877.0518	116.0245	0.0601	597.0332	397	236.3183	12,898.8203	116.1122	0.0751	598.0425
342	236.0850	13,030.2988	116.1367	0.0247	604.1384	398	236.1900	12,923.0303	116.0447	0.0474	599.1650
343	235.5800	12,961.0313	116.0235	-0.0839	600.9269	399	235.6567	12,906.2363	116.2321	-0.0676	598.3864
344	235.4650	12,864.4336	116.2333	-0.1091	596.4482	400	236.1117	12,933.4072	116.0829	0.0305	599.6461
345	236.4950	12,846.4629	116.2577	0.1135	595.6150	401	235.9633	12,952.9590	116.0888	-0.0014	600.5526
346	236.0717	12,919.7969	116.2511	0.0219	599.0151	402	236.1917	12,925.0850	116.1054	0.0478	599.2603
347	236.2167	12,924.6660	116.0915	0.0539	594.7638	403	235.7317	12,967.5908	116.2167	-0.0513	601.2310
348	235.9167	12,964.0498	116.0445	-0.0115	601.0668	404	235.7633	12,938.3535	116.1285	-0.0445	599.8754
349	236.0233	12,922.5879	116.0189	0.0115	599.1445	405	235.6083	12,833.8633	116.1015	-0.0782	595.0308
350	236.0567	12,915.4463	116.1498	0.0187	598.8134	406	235.5100	12,939.2266	116.1292	-0.0991	599.9159
351	235.9050	12,888.2100	116.0394	-0.0140	597.5506	407	236.1033	12,985.0195	116.0592	0.0287	602.0391
352	236.0517	12,862.2578	116.0609	0.0176	596.3473	408	236.0650	12,913.3408	116.1457	0.0205	598.7157
353	236.1033	12,924.6660	116.1440	0.0287	599.2408	409	235.8800	12,857.5537	116.2128	-0.0194	596.1292
354	235.6733	12,979.3408	116.0572	-0.0638	601.7758	410	236.3550	12,879.8643	116.1306	0.0831	597.1636
355	236.0517	12,999.7012	115.8885	0.0175	602.7198	411	235.5817	12,864.1455	116.1893	-0.0839	596.4349
356	236.2250	12,918.1826	116.1295	0.0550	598.9402	412	235.7500	12,865.6318	116.2078	-0.0475	596.5038
357	235.9000	12,913.5107	116.2255	-0.0151	598.7236	413	236.2667	12,871.2041	116.2221	0.0641	596.7621
358	235.7767	12,867.0977	115.9998	-0.0417	596.5717	414	236.1783	12,993.1143	116.1761	0.0448	602.4144
359	236.1517	12,921.6758	116.1354	0.0391	599.1022	415	235.9717	12,973.5830	116.1143	0.0004	601.5088
360	235.8200	12,908.2969	116.2142	-0.0323	598.4819	416	235.7216	12,943.9395	116.2542	-0.0535	600.1344
361	236.2533	12,937.5244	116.2033	0.0610	599.8370	417	236.0833	12,886.4883	116.2007	0.0245	597.4708
362	236.0017	12,891.4102	116.1797	0.0068	597.6990	418	236.1617	12,927.7344	116.0886	0.0413	599.3831
363	236.1117	12,937.2598	116.1739	0.0305	599.8247	419	235.7000	12,943.8154	115.9828	-0.0581	600.1287
364	235.9567	12,894.8135	116.2184	-0.0029	597.8567	420	236.0017	12,973.2969	116.0784	0.0068	601.4956
365	235.9467	12,962.4600	116.1193	-0.0050	600.9931	421	235.9533	12,918.7070	116.0212	-0.0036	598.9645
366	235.8867	12,897.7725	116.1145	-0.0180	597.9939	422	235.7783	12,950.2959	116.1510	-0.0413	600.4291
367	235.9250	12,849.1514	116.2921	-0.0097	595.7397	423	235.7733	13,008.1318	116.0481	-0.0422	603.1107
368	235.8000	12,902.7949	116.1851	-0.0367	598.2268	424	236.2934	12,939.7383	116.1478	0.0696	599.9396
369	235.9250	12,916.0049	116.2033	-0.0097	598.8393	425	235.9850	12,921.7676	116.1750	0.0032	599.1064
370	236.1000	12,980.9033	116.1833	0.0279	601.8482	426	235.9950	12,934.6328	116.1958	0.0054	599.7029
371	236.0533	12,940.6602	116.1092	0.0179	599.9824	427	235.9633	12,906.8428	116.0778	-0.0014	598.4145
372	236.0517	12,983.9297	116.1120	0.0176	601.9885	428	236.4400	12,972.5391	116.0422	0.1011	601.4604
373	236.0083	12,871.4072	116.2386	0.0083	596.7715	429	235.9267	12,902.3330	116.2049	-0.0093	598.2054
374	236.2017	12,867.5557	116.2061	0.0500	596.5930	430	235.7800	12,897.3926	116.0729	-0.0410	597.9763
375	235.7500	12,945.2793	116.1972	-0.0474	600.1965	431	235.7067	12,934.1748	116.1209	-0.0567	599.6817
376	236.2067	12,844.2627	116.1776	0.0512	595.5130	432	235.9367	12,923.5566	116.1593	-0.0072	599.1894
377	236.3900	12,999.2959	116.1643	0.0773	602.7010	433	236.3733	12,951.9365	116.0449	0.0868	600.5052
378	235.6817	12,942.7920	116.1346	-0.0621	600.0812	434	235.8783	12,963.9893	116.1045	-0.0197	601.0640
379	235.8817	12,905.1689	116.1372	-0.0190	598.3369	435	236.0183	12,944.0010	116.1518	0.0104	600.1373
380	235.7950	12,928.1924	116.1319	-0.0377	599.4043	436	236.0283	12,865.2764	116.2073	0.0126	596.4873
381	236.0650	12,921.4766	115.9760	0.0205	599.0930	437	235.7483	12,810.3369	116.2364	-0.0480	593.9401
382	236.1500	12,932.0635	116.1415	0.0388	599.5838	438	235.4100	12,901.9189	116.0402	-0.1208	598.1862
383	235.6100	12,867.2744	116.1303	-0.0777	596.5799	439	236.2083	12,903.0029	116.2509	0.0514	598.2364
384	235.7500	12,905.8955	116.2145	-0.0474	598.3706	440	236.2367	12,896.9600	116.0371	0.0575	597.9563
385	235.3267	12,925.4707	116.0578	-0.1386	599.2781	441	236.2250	12,935.0088	116.0689	0.0549	599.7204
386	235.4600	12,863.8574	116.0046	-0.1101	596.4215	442	235.7650	12,990.4541	116.1162	-0.0441	602.2910
387	235.9883	12,931.4717	116.1164	0.0039	599.5564	443	236.1067	12,804.6426	116.3210	0.0296	593.6760
388	236.0383	12,887.7266	116.0594	0.0147	597.5282	444	236.1917	12,962.1836	116.1501	0.0477	600.9803
389	235.5017	12,886.2041	116.1293	-0.1011	597.4576	445	235.8017	12,923.0723	116.2088	-0.0363	599.1669
390	236.1850	12,874.8887	116.1186	0.0464	596.9329	446	236.0517	12,908.3965	116.2170	0.0176	598.4865
391	235.6867	12,921.9863	116.1185	-0.0611	599.1166	447	236.0550	12,928.0049	116.1586	0.0183	599.3956
392	235.9883	12,955.5762	116.1881	0.0039	600.6740	448	235.9233	12,842.3975	116.3282	-0.0101	595.4265

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธี การสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12.919.4717 กรณีข้อสอบมีลักษณะการแจกแจงค่า

ความยาวของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
449	235.9233	12,956.4756	116.0731	-0.0100	600.7157	505	236.2167	12,847.1074	116.2680	0.0533	595.6449
450	236.1867	12,857.6318	116.2958	0.0468	596.1328	506	236.3783	12,949.6768	116.0071	0.0879	600.4004
451	236.0300	12,937.8281	116.1761	0.0129	599.8511	507	236.1417	13,027.2412	116.1229	0.0368	603.9966
452	235.8200	12,927.1904	116.1779	-0.0323	599.3579	508	236.0850	12,867.2988	116.2970	0.0248	596.5811
453	236.3400	13,024.7637	115.9679	0.0794	603.8818	509	236.1650	12,970.1055	116.1111	0.0419	601.3476
454	235.6566	12,903.0264	116.1322	-0.0676	598.2375	510	236.5117	13,022.9277	116.1062	0.1163	603.7966
455	235.7216	12,925.2012	116.1668	-0.0535	599.2656	511	236.0400	12,866.0459	116.2531	0.0151	596.5230
456	235.9450	12,922.0576	116.1886	-0.0054	599.1199	512	235.8567	12,953.0645	116.3125	-0.0244	600.5575
457	235.7817	12,866.1934	116.2245	-0.0407	596.5298	513	236.1383	12,889.6699	116.1609	0.0363	597.6183
458	236.6117	12,858.2646	116.2444	0.1386	596.1622	514	236.2650	12,897.0996	116.2547	0.0636	597.9627
459	235.4650	12,876.5742	116.0984	-0.1090	597.0111	515	235.9767	12,927.8047	116.2741	0.0014	599.3864
460	235.7917	12,948.5479	116.0748	-0.0384	600.3481	516	235.9050	12,896.3008	116.1894	-0.0140	597.9257
461	236.1467	12,865.3486	116.1421	0.0382	596.4906	517	235.8217	12,947.6455	116.0871	-0.0319	600.3063
462	235.7333	12,858.1201	116.2358	-0.0511	596.1555	518	235.9667	12,851.8252	116.2368	-0.0007	595.8636
463	235.9733	12,935.6396	116.0780	0.0007	599.7496	519	235.7400	12,887.9551	116.2480	-0.0496	597.5388
464	235.7867	12,955.9707	116.0824	-0.0395	600.6922	520	236.0967	12,905.9072	116.2569	0.0273	598.3711
465	236.0950	12,873.2217	116.3047	0.0270	596.8557	521	235.7550	12,887.9092	116.3172	-0.0464	597.5366
466	235.9767	12,936.9785	116.1699	0.0014	599.8117	522	235.8600	13,032.5420	116.0584	-0.0236	604.2424
467	236.0117	12,955.1260	116.1979	0.0090	600.6531	523	236.1100	12,825.7617	116.2513	0.0303	594.6552
468	235.9200	12,990.8818	116.0793	-0.0107	602.3109	524	236.3683	12,949.3906	116.1615	0.0857	600.3872
469	236.1367	12,876.2920	115.9935	0.0360	596.9980	525	235.8567	12,881.1973	116.0108	-0.0245	597.2254
470	235.8467	12,879.3223	116.2738	-0.0266	597.1385	526	235.9300	12,970.4180	116.0834	-0.0086	601.3621
471	236.0083	12,917.5439	116.1980	0.0083	598.9106	527	236.1483	12,883.8008	116.1851	0.0385	597.3462
472	235.3967	12,976.9785	116.0813	-0.1233	601.6663	528	236.0650	12,904.5635	116.1378	0.0205	598.3088
473	235.6200	12,913.0781	116.1871	-0.0754	598.7036	529	235.9233	12,960.3838	116.2034	-0.0100	600.8969
474	235.9900	12,886.5127	116.1540	0.0043	597.4719	530	236.0467	12,846.3818	116.3176	0.0166	595.6113
475	236.3050	12,883.1611	116.2082	0.0723	597.3165	531	235.6900	12,953.4092	116.2711	-0.0603	600.5735
476	235.9633	12,931.6455	116.1912	-0.0014	599.5644	532	235.8283	12,837.6807	116.3083	-0.0306	595.2078
477	236.0167	12,930.1582	116.1560	0.0101	599.4955	533	235.8633	12,980.5723	116.1336	-0.0229	601.8329
478	235.8767	12,942.4883	116.1952	-0.0201	600.0671	534	236.1450	12,901.1270	116.1755	0.0377	598.1495
479	236.3567	12,889.7236	116.1731	0.0834	597.6208	535	235.7383	12,892.7754	116.2919	-0.0500	597.7622
480	236.4600	12,938.6621	116.0467	0.1055	599.8897	536	235.7267	12,918.0684	116.1600	-0.0524	598.9349
481	236.1917	12,960.0713	116.1371	0.0477	600.8824	537	236.3350	12,915.4004	116.2161	0.0787	598.8112
482	236.0200	12,915.9570	116.1031	0.0108	598.8370	538	235.6250	13,043.8496	116.0342	-0.0740	604.7667
483	236.2617	12,890.5010	116.1463	0.0629	597.6568	539	235.5500	12,888.6465	116.1414	-0.0906	597.5708
484	235.6067	12,952.9131	116.0666	-0.0782	600.5505	540	235.9067	12,841.3770	116.2134	-0.0137	595.3792
485	235.8783	12,926.2529	116.2392	-0.0197	599.3144	541	236.0617	12,931.2617	116.2774	0.0197	599.5466
486	236.0950	12,841.1309	116.2108	0.0270	595.3678	542	235.7117	12,895.4736	116.1292	-0.0557	597.8874
487	235.7833	12,948.5977	116.1690	-0.0402	600.3504	543	235.8950	12,912.4209	116.2302	-0.0162	598.6731
488	236.0867	12,953.4355	116.0203	0.0251	600.5747	544	235.9883	12,915.2549	115.9866	0.0039	598.8045
489	235.6850	12,874.1465	116.2800	-0.0615	596.8985	545	235.6883	12,926.4365	116.1683	-0.0607	599.3229
490	235.4850	12,834.5107	116.1739	-0.1049	595.0609	546	235.7800	12,966.9629	116.0780	-0.0409	601.2019
491	235.9950	12,897.4521	116.2320	0.0054	597.9791	547	235.9317	12,911.9980	116.1037	-0.0083	598.6535
492	235.5867	12,930.7070	116.3723	-0.0826	599.5209	548	235.8600	12,951.0869	116.2932	-0.0237	600.4658
493	236.4133	12,803.9854	116.1534	0.0960	593.6456	549	236.2267	12,852.2217	116.1582	0.0555	595.8820
494	235.8517	12,952.6182	116.1290	-0.0255	600.5368	550	236.3150	12,793.0283	116.2095	0.0747	593.1376
495	235.9267	12,872.6973	116.2065	-0.0094	596.8313	551	236.2200	13,065.1729	116.0765	0.0536	605.7553
496	236.1100	12,924.3965	116.0590	0.0302	599.2283	552	236.0233	12,990.3340	115.9999	0.0115	602.2855
497	236.3100	12,889.1914	116.1702	0.0734	597.5961	553	235.8467	12,905.4834	116.2277	-0.0266	598.3514
498	236.3617	12,930.8467	116.1403	0.0844	599.5274	554	235.8117	12,941.7969	116.3020	-0.0341	600.0351
499	236.4233	12,838.4922	116.2945	0.0980	595.2455	555	235.8183	12,862.5078	116.3480	-0.0328	596.3589
500	236.1433	12,900.8281	116.0782	0.0374	598.1356	556	235.9283	12,877.2969	116.1160	-0.0090	597.0446
501	236.2233	12,915.9961	116.1878	0.0546	598.8389	557	235.6283	12,878.2881	116.2025	-0.0737	597.0906
502	236.0967	12,927.4941	116.3110	0.0273	599.3720	558	235.6983	13,042.9434	116.0026	-0.0583	604.7247
503	236.2433	12,853.6172	116.1687	0.0591	595.9467	559	235.9333	12,835.2275	116.2506	-0.0079	595.0941
504	235.8150	12,936.7031	116.1068	-0.0334	599.7989	560	235.8933	12,981.5059	116.1451	-0.0165	601.8762

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 236.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีลักษณะการแจกแจงค่า

ความยาวของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
561	235.8583	12,886.1680	116.2086	-0.0241	597.4559	617	235.8517	12,897.0869	116.1622	-0.0255	597.9621
562	235.6233	12,975.7598	115.9858	-0.0745	601.6097	618	235.6350	12,841.1211	116.2525	-0.0724	595.3673
563	236.2300	12,865.9717	116.2216	0.0561	596.5195	619	236.0750	12,860.8496	116.2353	0.0227	596.2820
564	235.8483	12,931.4746	116.2188	-0.0262	599.5565	620	236.0700	12,963.7539	116.1048	0.0215	601.0531
565	235.9383	12,950.0771	115.9191	-0.0068	600.4190	621	235.9800	12,873.4854	116.1023	0.0022	596.8679
566	236.1250	12,953.3564	116.1364	0.0334	600.5710	622	236.1350	12,875.1934	116.1324	0.0356	596.9471
567	235.6067	12,945.4063	116.1410	-0.0782	600.2024	623	235.8683	12,960.5176	116.0553	-0.0219	600.9031
568	235.5067	12,975.0176	116.0413	-0.0996	601.5753	624	235.2867	12,899.7852	116.1304	0.0683	598.0873
569	236.3250	12,921.9512	116.2374	0.0765	599.1150	625	236.2917	12,939.0527	116.1054	0.0693	599.9079
570	235.9750	12,817.3252	116.1709	0.0011	594.2641	626	235.7217	12,895.8057	116.1110	-0.0536	597.9027
571	235.9500	12,938.7031	116.0705	-0.0043	599.8916	627	236.4483	12,854.0620	116.1956	0.1033	595.9683
572	235.4183	12,946.8184	116.0074	-0.1188	600.2679	628	235.8867	13,006.3848	116.0904	-0.0179	603.0296
573	236.0050	12,897.2939	116.1191	0.0075	597.9717	629	235.6633	13,018.0557	115.9898	-0.0658	603.5708
574	235.8467	12,944.5273	116.1699	-0.0266	600.1617	630	236.0700	12,865.3555	116.2660	0.0216	596.4909
575	235.9767	12,903.3633	116.2096	0.0014	598.2531	631	235.7817	12,881.5146	116.1623	-0.0406	597.2402
576	235.7833	13,006.5234	116.0147	-0.0401	603.0361	632	236.1100	12,834.7256	116.3165	0.0303	595.0708
577	236.1900	12,898.1826	116.1617	0.0475	598.0130	633	236.1100	12,916.6816	116.2089	0.0302	598.8706
578	235.9000	12,909.6133	116.1611	-0.0151	598.5429	634	236.1033	12,861.3027	116.2663	0.0288	596.3030
579	235.6400	12,890.6748	116.2089	-0.0712	597.6649	635	235.7350	12,864.0996	116.1350	-0.0508	596.4327
580	236.0450	12,764.8262	116.3873	0.0163	591.8900	636	236.1633	12,951.4971	116.1028	0.0416	600.4848
581	236.2267	12,892.1904	116.2738	0.0554	597.7351	637	235.8017	12,841.0537	116.3700	-0.0364	595.3642
582	236.0617	12,931.2188	116.1860	0.0197	599.5446	638	236.1400	13,001.7070	116.2251	0.0365	602.8128
583	236.4100	12,896.7354	116.2776	0.0949	597.9459	639	236.3333	12,887.8662	116.1758	0.0784	597.5346
584	235.6833	12,935.0850	116.0406	-0.0617	599.7239	640	236.1650	12,967.8340	116.0316	0.0419	601.2423
585	235.8000	12,911.8633	116.0636	-0.0366	598.6472	641	235.6283	12,994.3252	116.1833	-0.0734	602.4705
586	236.0383	12,977.7803	116.1073	0.0147	601.7034	642	236.1383	12,955.7549	116.0515	0.0362	600.6822
587	235.7800	12,969.6670	116.1441	-0.0409	601.3273	643	236.4733	12,904.6582	116.2959	0.1085	598.3132
588	236.0817	12,976.8975	116.1001	0.0240	601.6625	644	236.0933	12,978.8965	116.0052	0.0265	601.7552
589	235.4367	12,868.8457	116.2603	-0.1152	596.6528	645	236.3300	12,936.1523	116.0755	0.0775	599.7734
590	235.7500	12,944.0781	116.1107	-0.0474	600.1409	646	236.3100	13,019.8867	116.1040	0.0730	603.6557
591	235.4133	12,935.5020	116.0835	-0.1199	599.7432	647	236.2334	12,954.8652	116.3905	0.0567	600.6410
592	235.6967	12,929.5635	116.0500	-0.0589	599.4679	648	235.5883	12,939.3252	116.2359	-0.0822	599.9205
593	235.8283	12,837.3379	116.1971	-0.0306	595.1919	649	235.9750	12,953.2266	116.1568	0.0011	600.5650
594	235.9633	12,905.7959	116.2063	-0.0014	598.3659	650	236.5067	12,932.6045	115.9244	0.1156	599.6089
595	236.2150	12,891.4258	116.1388	0.0529	597.6997	651	235.8567	12,943.6650	116.0640	-0.0244	600.1217
596	236.1767	12,937.2744	116.1933	0.0445	599.8254	652	236.0883	12,906.5576	116.1474	0.0255	598.4013
597	235.8167	12,854.4043	116.2367	-0.0331	595.9832	653	235.7983	12,995.5303	116.1356	-0.0369	602.5264
598	235.7500	12,977.1875	116.1692	-0.0473	601.6759	654	235.8433	12,861.8359	116.2184	-0.0274	596.3278
599	236.0750	12,918.1914	116.1972	0.0226	598.9406	655	235.9350	12,923.5156	116.1266	-0.0075	599.1875
600	236.0367	12,795.7061	116.2933	0.0144	593.2617	656	236.3417	12,839.3682	116.1744	0.0803	595.2861
601	235.6400	12,939.1797	116.1313	-0.0711	599.9137	657	235.6500	12,905.9590	116.1615	-0.0690	598.3735
602	235.9650	12,894.6104	116.0647	-0.0011	597.8473	658	236.4767	12,886.2793	116.3048	0.1093	597.4611
603	235.7833	12,814.0820	116.2047	-0.0404	594.1137	659	236.0083	12,914.4570	116.2133	0.0083	598.7675
604	235.9383	12,871.2021	116.2892	-0.0068	596.7620	660	235.8383	12,881.9365	116.1384	-0.0284	597.2597
605	236.0617	13,031.2852	116.0457	0.0197	604.1841	661	236.0600	12,845.7852	116.1825	0.0195	595.5836
606	235.8700	13,006.9600	116.0678	-0.0215	603.0563	662	236.0033	12,911.4814	116.1193	0.0072	598.6295
607	236.1383	12,996.5908	116.1280	0.0362	602.5756	663	236.0117	12,882.0703	116.0275	0.0090	597.2659
608	236.3567	12,910.0752	116.0371	0.0834	598.5643	664	235.9383	12,910.3447	116.1414	-0.0068	598.5768
609	236.1267	12,964.1758	116.1236	0.0337	601.0727	665	236.1567	12,978.9346	116.0764	0.0401	601.7569
610	236.1267	12,932.6338	116.3345	0.0337	599.6102	666	235.5133	12,894.9229	116.1555	-0.0985	597.8618
611	235.7983	12,878.9775	116.2495	-0.0371	597.1225	667	235.9417	12,927.2471	116.2001	-0.0061	599.3605
612	235.4933	12,885.0254	116.1526	-0.1029	597.4029	668	236.1050	12,975.2607	116.2489	0.0290	601.5866
613	236.0950	12,877.9746	116.1923	0.0270	597.0760	669	236.2500	12,894.6709	116.0610	0.0604	597.8501
614	235.7000	12,897.3740	116.2430	-0.0582	597.9755	670	236.3800	12,994.5332	116.1581	0.0881	602.4802
615	236.0083	12,888.7363	116.1952	0.0083	597.5750	671	236.3917	12,917.6816	116.1528	0.0909	598.9170
616	235.9033	12,926.3965	116.1382	-0.0144	599.3211	672	236.2467	12,986.3711	116.2346	0.0595	602.1017

ตาราง 23 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่ผู้สมัครได้จากการสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีนี้ข้อสอบมีลักษณะการแจกแจงค่า

ความยากของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
673	236.0033	12,857.7354	116.2216	0.0072	596.1376	729	235.8783	12,888.7529	116.3107	-0.0198	597.5758
674	235.7717	12,929.2891	116.1504	-0.0427	599.4552	730	235.6567	12,928.6416	116.0349	-0.0675	599.4252
675	236.0300	12,986.8682	116.0820	0.0129	602.1248	731	235.9600	12,913.5889	116.0990	-0.0022	598.7272
676	236.1167	12,951.5645	116.0402	0.0316	600.4880	732	236.0100	12,890.9678	116.2024	0.0086	597.6784
677	236.0800	12,903.3301	116.0452	0.0237	598.2516	733	235.6300	12,999.4268	116.1328	-0.0730	602.7070
678	235.6033	12,924.2871	116.1350	-0.0790	599.2233	734	235.9800	12,962.0371	116.0413	0.0022	600.9735
679	236.0733	12,912.8389	116.0839	0.0223	598.6925	735	235.8150	12,896.2461	116.1664	-0.0334	597.9232
680	235.8083	12,846.2246	116.2087	-0.0349	595.6040	736	236.4750	12,908.8291	116.2727	0.1089	598.5066
681	236.1783	12,928.7246	116.1034	0.0449	599.4290	737	236.1717	12,872.3838	116.3432	0.0435	596.8168
682	235.4683	12,981.5859	116.1209	-0.1079	601.8799	738	236.2350	12,907.7617	116.2418	0.0571	598.4571
683	236.3133	12,880.0908	116.1436	0.0741	597.1741	739	235.9717	12,927.0205	116.3754	0.0004	599.3500
684	236.2750	12,944.1670	116.1857	0.0657	600.1450	740	235.9583	12,884.0527	116.1492	-0.0025	597.3578
685	235.8550	12,891.4072	116.1738	-0.0248	597.6988	741	235.5917	13,029.7275	116.0878	-0.0812	604.1119
686	235.8650	12,891.4160	116.2467	-0.0227	597.6992	742	236.1483	12,933.7900	116.1317	0.0384	599.6639
687	235.7700	13,010.2617	116.1142	-0.0429	603.2094	743	236.1300	12,896.5420	116.1235	0.0345	597.9369
688	236.1167	12,903.4678	116.3496	0.0316	598.2580	744	236.0817	12,893.7441	116.2464	0.0241	597.8072
689	235.4833	12,854.4678	116.1715	-0.1051	595.9862	745	236.1667	12,871.1846	116.0958	0.0425	596.7612
690	236.1167	12,963.4824	116.1809	0.0316	601.0405	746	235.9833	12,882.3291	116.0179	0.0029	597.2779
691	235.8283	12,832.3008	116.1629	-0.0306	594.9584	747	236.0133	12,798.6094	116.2436	0.0094	593.3963
692	235.8633	12,975.1309	116.1173	-0.0229	601.5806	748	236.2900	12,893.2598	116.2390	0.0690	597.7847
693	235.5633	12,956.6934	116.1314	-0.0875	600.7258	749	236.1117	12,937.3555	116.1397	0.0305	599.8292
694	235.5750	12,964.5381	116.1459	-0.0850	601.0895	750	236.4100	12,918.0654	116.1628	0.0948	598.9348
695	236.2383	12,989.0830	116.1097	0.0577	602.2275	751	236.1917	12,945.7949	116.1486	0.0477	600.2205
696	235.9700	12,931.1895	116.1204	0.0000	599.5433	752	236.0950	12,834.2607	116.1879	0.0270	595.0493
697	236.0150	13,023.3086	116.1827	0.0097	603.8143	753	235.8650	12,884.6621	116.1729	-0.0227	597.3861
698	235.8817	12,955.3027	115.9790	-0.0190	600.6613	754	236.3100	12,938.2344	116.0943	0.0732	599.8639
699	235.8183	12,835.9844	116.1597	-0.0328	595.1292	755	236.5533	12,927.4531	116.2075	0.1257	599.3701
700	235.7150	12,877.6055	116.1992	-0.0550	597.0589	756	235.5450	12,916.7324	116.2244	-0.0916	598.8730
701	235.6400	12,872.6133	116.1681	-0.0712	596.8275	757	235.7900	12,904.4453	116.1627	-0.0388	598.3033
702	236.3483	12,861.9590	116.0963	0.0817	596.3335	758	235.8917	12,985.6357	116.0725	-0.0168	602.0676
703	236.1383	12,910.2725	116.3719	0.0363	598.5735	759	236.1767	12,895.4209	116.2999	0.0446	597.8849
704	236.1017	12,894.4238	116.0171	0.0284	597.8387	760	236.1933	12,867.8721	116.3650	0.0482	596.6076
705	235.9533	12,835.5889	116.1164	-0.0036	595.1108	761	236.0950	12,913.3027	116.1335	0.0269	598.7140
706	235.8950	12,928.0293	116.1043	0.0054	599.3968	762	235.9317	12,986.9043	116.1154	-0.0082	602.1265
707	236.1200	12,911.4463	116.2508	0.0323	598.6279	763	236.4333	12,942.4287	116.1070	0.0998	600.0644
708	236.0433	12,926.1719	116.1218	0.0158	599.3106	764	236.0050	12,925.4961	116.0550	0.0075	599.2793
709	235.9567	12,855.8818	116.3146	-0.0029	596.0517	765	235.9733	12,950.6543	116.1391	0.0007	600.4458
710	235.5217	12,919.4414	116.2119	-0.0966	598.9986	766	235.7583	12,844.3018	116.2535	-0.0457	595.5148
711	235.8817	12,926.1709	116.1265	-0.0190	599.3106	767	235.7067	12,908.2158	116.1971	-0.0568	598.4781
712	235.5833	12,924.2314	116.0614	-0.0833	599.2207	768	236.3567	12,960.6836	116.1077	0.0832	600.9108
713	236.3800	12,836.6826	116.0819	0.0886	595.1616	769	235.6800	12,947.3066	116.0688	-0.0624	600.2905
714	236.2450	12,912.6191	116.1348	0.0593	598.6823	770	235.6000	13,009.9482	116.0617	-0.0795	603.1949
715	235.6383	12,930.5547	116.0861	-0.0714	599.5139	771	235.9917	12,901.6162	116.0464	0.0047	598.1721
716	235.9333	12,912.0508	116.1964	-0.0079	598.6559	772	236.0200	12,939.7295	116.1468	0.0108	599.9392
717	235.9017	12,923.5479	116.2973	-0.0147	599.1890	773	236.1183	12,968.9600	116.0259	0.0319	601.2945
718	236.1267	12,925.6191	116.1719	0.0338	599.2850	774	235.8617	12,911.0352	116.1839	-0.0234	598.6088
719	236.1150	12,917.7744	116.2176	0.0313	598.9213	775	235.2417	12,893.0811	116.0808	-0.1571	597.7764
720	235.8250	13,003.9160	116.1038	-0.0311	602.9152	776	235.9150	12,843.3564	116.3266	-0.0119	595.4710
721	235.7017	12,982.2461	116.0372	-0.0577	601.9105	777	235.7867	12,955.7334	116.0106	-0.0395	600.6812
722	235.9750	12,890.7588	116.1397	0.0011	597.6688	778	236.2917	12,910.3154	116.2415	0.0693	598.5755
723	236.2967	12,885.0322	116.2479	0.0705	597.4032	779	236.0717	12,909.3301	116.1320	0.0219	598.5298
724	236.5867	12,921.1318	116.1050	0.1329	599.0770	780	236.1733	12,946.6787	116.0819	0.0438	600.2614
725	235.4767	12,898.2373	116.1661	-0.1064	598.0155	781	235.9800	12,968.6758	116.1367	0.0022	601.2813
726	236.2983	12,974.6270	116.1222	0.0706	601.5572	782	235.9433	12,903.7207	116.1918	-0.0057	598.2697
727	235.9017	12,944.9814	116.0459	-0.0147	600.1827	783	236.2417	12,961.0781	116.1293	0.0585	600.9290
728	236.1100	12,946.5127	116.1485	0.0301	600.2537	784	235.8500	12,885.8252	115.9392	-0.0259	597.4400

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่สุ่มโดยวิธีการสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร

ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีข้อสอบมีลักษณะการแจกแจงค่า

ความยาวของข้อสอบแบบปกติ จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน

No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
785	235.7250	12,949.8711	116.2374	-0.0527	600.4094	841	235.8883	12,945.3184	116.0439	-0.0176	600.1984
786	236.1133	12,894.0078	116.2205	0.0309	597.8194	842	235.5950	13,031.7256	116.1119	-0.0805	604.2046
787	236.2633	12,906.8408	116.2046	0.0632	598.4144	843	236.2150	12,903.9688	116.1427	0.0528	598.2812
788	236.3683	12,876.3389	116.2725	0.0860	597.0002	844	236.2200	13,027.1484	116.0727	0.0537	603.9923
789	236.3317	12,901.8926	116.2019	0.0780	598.1850	845	235.9600	12,898.6787	116.0587	-0.0022	598.0360
790	235.9984	12,961.9414	116.0455	0.0061	600.9691	846	235.9033	12,851.2070	115.9898	-0.0144	595.8350
791	235.8533	12,925.5537	116.1568	-0.0251	599.2820	847	235.9000	12,941.7256	116.0775	-0.0151	600.0318
792	235.2200	12,807.5557	116.3255	-0.1623	593.8111	848	235.9167	12,871.3271	116.2656	-0.0115	596.7678
793	235.8400	12,970.3184	116.2647	-0.0280	601.3575	849	236.1200	12,891.1768	116.0777	0.0324	597.6881
794	236.0267	12,948.9785	116.1743	0.0122	600.3681	850	236.0067	12,887.1240	116.1775	0.0079	597.5002
795	236.0650	12,943.5283	116.1378	0.0205	600.1154	851	236.0633	12,881.1260	116.2196	0.0201	597.2221
796	236.0283	12,817.5420	116.0827	0.0126	594.2741	852	235.9283	12,762.4297	116.1067	-0.0090	591.7189
797	235.7083	12,950.6914	116.1692	-0.0563	600.4475	853	235.7767	12,939.9102	116.0940	-0.0416	599.9476
798	236.2333	12,972.7764	116.1585	0.0566	601.4714	854	236.3067	12,924.3271	116.2282	0.0725	599.2251
799	236.1700	12,879.0234	116.1970	0.0432	597.1247	855	236.0933	12,931.0283	116.1660	0.0266	599.5358
800	236.2783	12,808.9707	116.1696	0.0667	593.8767	856	236.0100	12,902.1328	116.2961	0.0086	598.1961
801	236.0867	12,959.1670	115.9370	0.0251	600.8404	857	236.2200	12,904.7002	116.1872	0.0539	598.3151
802	236.0050	12,863.2949	116.3195	0.0076	596.3954	858	235.5967	12,879.1689	116.2450	-0.0806	597.1314
803	236.5200	12,895.1357	116.1988	0.1186	597.8717	859	235.9050	12,948.4033	116.1495	-0.0140	600.3414
804	235.7500	12,962.7930	116.1907	-0.0473	601.0086	860	235.9733	12,908.8291	116.1773	0.0007	599.5066
805	236.2383	12,772.6191	116.3187	0.0582	592.1913	861	236.0233	13,015.1582	116.1763	0.0114	603.4364
806	236.1300	12,911.6963	116.1740	0.0345	598.6395	862	235.8067	12,968.8398	116.1489	-0.0351	601.2889
807	236.0067	12,938.3398	116.0573	0.0079	599.8748	863	236.5183	13,027.6719	116.0390	0.1177	604.0166
808	235.7950	12,968.0039	116.1060	-0.0376	601.2502	864	236.1167	12,875.9580	116.0508	0.0317	596.9825
809	236.0967	12,950.5645	116.2314	0.0273	600.4416	865	236.1133	12,898.3281	116.2081	0.0309	598.0197
810	235.7683	12,892.9092	116.2779	-0.0435	597.7685	866	236.0150	13,005.1035	116.2791	0.0097	602.9702
811	236.2367	12,896.1885	116.1468	0.0575	597.9205	867	236.0650	12,875.4375	116.1426	0.0205	596.9584
812	235.9467	12,989.8535	116.1216	-0.0050	602.2632	868	236.1183	13,031.1777	116.0275	0.0318	604.1792
813	236.2850	12,985.0947	116.1166	0.0677	602.0426	869	235.8300	12,917.1943	116.1708	-0.0302	598.8944
814	236.2383	12,915.9844	116.1289	0.0578	598.8383	870	235.8533	12,863.9053	116.1940	-0.0252	596.4237
815	236.0733	12,957.8301	116.1735	0.0222	600.7785	871	235.4917	12,879.4717	116.0757	-0.1032	597.1454
816	235.5650	12,954.5908	116.1355	-0.0872	600.6283	872	235.8517	12,970.8848	116.1499	-0.0255	601.3837
817	235.4433	12,781.7813	116.1555	-0.1141	592.6161	873	235.7250	12,978.9082	116.1794	-0.0527	601.7557
818	235.6417	12,962.9600	116.2206	-0.0706	601.0163	874	235.7633	12,998.1758	116.2173	-0.0444	602.6490
819	236.1817	12,898.7900	116.0188	0.0457	598.0411	875	235.6533	12,831.7100	116.3071	-0.0685	594.9310
820	236.4083	12,888.3340	116.1678	0.0946	597.5563	876	236.0700	12,911.8789	116.1318	0.0216	598.6480
821	235.6200	12,955.2637	116.0174	-0.0753	600.6595	877	236.6300	12,949.6738	116.0405	0.1421	600.4003
822	236.0683	12,855.2676	116.1158	0.0212	596.0232	878	235.8133	12,933.5352	116.1231	-0.0337	599.6520
823	236.0733	12,935.4805	116.2559	0.0223	599.7422	879	236.3233	12,947.2852	116.1129	0.0761	600.2895
824	235.8050	12,906.7617	116.2070	-0.0356	598.4107	880	235.7267	12,957.2148	115.9756	-0.0524	600.7499
825	236.3633	12,891.1553	116.2082	0.0849	598.6871	881	235.8883	12,866.8291	116.3004	-0.0176	596.5593
826	235.6417	13,026.5283	115.9707	-0.0705	603.9636	882	236.4167	12,946.0020	116.2477	0.0962	600.2301
827	236.0717	12,929.0303	116.1705	0.0219	599.4432	883	236.3900	12,986.4922	116.1242	0.0903	602.1073
828	236.2267	12,952.2168	116.0322	0.0552	600.5182	884	235.8550	12,827.0098	116.1709	-0.0249	594.7131
829	235.8567	12,939.5879	116.1924	-0.0244	599.9327	885	235.9250	12,915.0898	116.1395	-0.0097	598.7968
830	235.8967	12,909.9648	116.2109	-0.0158	598.5592	886	236.0350	12,847.6631	116.1204	0.0140	595.6707
831	235.9767	13,031.2256	116.0461	0.0014	604.1814	887	236.0650	12,885.0029	116.1132	0.0205	597.4019
832	236.3150	12,910.2354	116.2373	0.0744	598.5718	888	236.1250	12,939.3477	116.0793	0.0334	599.9215
833	235.7817	12,907.6123	116.0968	-0.0406	598.4502	889	236.0750	12,909.8828	116.1977	0.0226	598.5554
834	236.3850	12,969.1514	116.0853	0.0850	601.3034	890	235.9800	12,899.2852	116.1356	0.0022	598.0641
835	236.0783	12,966.3633	116.1512	0.0233	601.1741	891	236.2483	12,874.1602	116.2167	0.0601	596.8992
836	235.9117	12,982.9932	116.0841	-0.0125	601.9451	892	235.8967	12,981.4824	116.1596	-0.0158	601.8751
837	236.1567	13,000.6221	116.1285	0.0401	602.7625	893	235.9350	12,961.2070	116.0722	-0.0075	600.9350
838	236.4033	12,939.6455	116.1496	0.0933	599.9353	894	236.0850	12,818.4551	116.1006	0.0249	594.3165
839	236.2583	13,008.8828	116.0012	0.0619	603.1455	895	236.0750	12,931.0439	116.2777	0.0226	599.5365
840	236.2133	12,799.5576	116.3738	0.0527	593.4403	896	235.8400	12,870.5957	116.2184	-0.0281	596.7339

ตาราง 29 (ต่อ) เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนคะแนนที่ผู้วิจัยใช้การสุ่มแบบเมตริกซ์เชิงพหุคูณ กับค่าคะแนนเฉลี่ยของประชากร
ซึ่งเท่ากับ 235.9700 และความแปรปรวนคะแนนของประชากรซึ่งเท่ากับ 12,919.4717 กรณีที่ข้อสอบมีลักษณะการแจกแจงค่า

ความยากของข้อสอบแบบปกติ						จำนวนตัวอย่างผู้สอบเท่ากับ 600 คน					
No.	Mean	Variance	S.E.Mean	t-value	Chi-square	No.	Mean	Variance	S.E.Mean	t-value	Chi-square
897	235.9133	12,931.8916	116.1665	-0.0122	599.5758	953	236.2533	12,976.9902	116.0967	0.0609	601.6668
898	235.4350	12,857.5098	116.1590	-0.1156	596.1272	954	235.7017	12,844.3818	116.2805	-0.0580	595.5185
899	235.6967	12,876.3682	116.2382	-0.0590	597.0015	955	235.9533	12,867.8262	116.0770	-0.0036	601.2419
900	236.2683	12,800.6943	116.2203	0.0646	593.4930	956	236.0600	12,908.4219	116.0825	0.0194	598.4877
901	235.5300	12,913.5479	116.1776	-0.0948	598.7253	957	235.8767	12,915.6436	115.9904	-0.0201	598.8225
902	235.7400	12,753.8604	116.2449	-0.0499	591.3216	958	235.8817	12,883.7632	116.1284	-0.0191	597.3453
903	235.7033	12,859.7705	116.1159	-0.0576	596.2320	959	235.7150	12,926.3555	116.1190	-0.0549	599.3192
904	236.1933	12,911.3379	116.1933	0.0481	598.6229	960	236.0183	12,813.4785	116.1314	0.0105	594.0857
905	236.1783	12,944.6377	116.0549	0.0449	600.1668	961	236.1567	13,002.8047	116.0307	0.0401	602.8637
906	236.0150	13,060.9180	115.9467	0.0096	605.5580	962	235.8833	12,893.2412	116.2622	-0.0187	597.7838
907	235.6400	12,933.6406	115.9966	-0.0711	599.6569	963	235.5700	12,930.0010	116.2593	-0.0862	599.4882
908	236.5917	13,009.0508	115.9894	0.1335	603.1593	964	236.3267	12,973.9570	116.1285	0.0767	601.5262
909	235.9483	12,864.3496	116.2454	-0.0047	596.4443	965	235.6067	13,045.0674	116.0948	-0.0779	604.8231
910	235.7000	12,999.7979	116.1484	-0.0580	602.7243	966	235.6700	12,938.1660	115.9604	-0.0648	599.8667
911	236.1683	12,896.7715	116.1541	0.0428	597.9475	967	235.2767	12,994.1357	116.0973	-0.1490	602.4617
912	236.2483	12,887.2480	116.2709	0.0601	597.5060	968	235.7867	12,948.2539	116.1659	-0.0395	600.3345
913	236.3050	12,876.9014	116.1326	-0.1435	597.0263	969	236.2100	12,884.1602	116.2391	0.0518	597.3628
914	236.0000	12,962.3105	116.0124	0.0065	600.9862	970	236.0467	12,931.4307	116.3187	0.0165	599.5545
915	236.0650	13,040.3340	116.0731	0.0204	604.6037	971	236.0550	12,861.7861	116.2454	0.0184	596.3255
916	235.6000	12,950.1943	115.9771	-0.0796	600.4244	972	236.2067	12,889.7832	116.1647	0.0511	597.6235
917	235.7267	12,952.2148	115.9899	-0.0524	600.5181	973	236.0083	12,947.1084	116.1705	0.0083	600.2814
918	235.8983	12,943.7275	116.1047	-0.0154	600.1246	974	235.8783	12,911.8662	116.2100	-0.0198	598.6474
919	235.8933	12,875.5488	116.0566	-0.0165	596.9636	975	236.0533	12,965.8896	116.1137	0.0179	601.1521
920	235.8633	12,876.1094	116.1500	-0.0230	596.8895	976	235.7333	12,978.7598	116.0146	-0.0509	601.7488
921	235.8733	13,018.4297	116.1115	-0.0208	603.5881	977	235.6400	12,829.6992	116.1924	-0.0714	594.8378
922	236.0033	12,825.5117	116.2569	0.0072	594.6436	978	235.8717	12,907.5928	116.1835	-0.0212	598.4492
923	235.9867	12,939.0811	116.0931	0.0036	599.9092	979	235.9800	12,916.8008	116.0689	0.0022	598.8762
924	235.8267	12,910.7461	116.0844	-0.0309	598.5954	980	236.0467	12,983.2773	116.1702	0.0165	601.9583
925	235.8917	12,875.0654	116.2868	-0.0169	596.9411	981	235.6083	12,944.7148	116.2360	-0.0779	600.1704
926	235.9367	12,941.2402	116.0730	-0.0072	600.0093	982	235.8667	12,901.9131	116.1285	-0.0223	598.1859
927	236.0050	12,964.8008	116.1991	0.0075	601.1016	983	236.2450	12,883.2666	116.1394	0.0593	597.3214
928	236.3467	12,931.1455	116.0995	0.0811	599.5412	984	236.0217	12,895.4775	116.1449	0.0111	597.8875
929	236.0067	12,952.4258	116.0199	0.0079	600.5279	985	236.0683	12,853.2002	116.1976	0.0212	595.8274
930	235.6750	12,852.5684	116.0356	-0.0637	595.8981	986	235.8967	12,941.1455	116.2705	-0.0158	600.0049
931	235.5783	12,957.7354	116.1371	-0.0843	600.7741	987	235.9900	12,872.6025	116.1578	0.0043	596.8270
932	235.9967	12,921.8965	116.1967	0.0057	599.1124	988	235.2433	12,884.0762	116.2016	-0.1568	597.3589
933	235.9800	12,986.0137	116.0097	0.0022	602.0852	989	236.2217	12,889.0645	116.2620	0.0543	597.5902
934	236.7893	12,978.4434	116.0447	0.1760	601.7342	990	235.7800	12,922.9189	115.9852	-0.0409	599.1598
935	235.6283	12,920.9863	116.1677	-0.0305	599.0702	991	236.1100	12,940.8438	116.1196	0.0301	599.9909
936	235.6567	12,884.8105	116.1739	-0.0676	597.9930	992	236.1633	12,973.4131	116.1093	0.0416	601.5009
937	236.3933	12,982.2314	116.2219	0.0910	601.9098	993	235.8667	12,866.5459	116.1169	-0.0223	596.5461
938	236.0317	12,702.5879	116.2297	0.0134	588.9444	994	235.9217	12,899.3027	116.1191	-0.0104	598.0649
939	236.0067	12,858.4590	116.2263	0.0079	596.1712	995	235.9883	12,961.0566	116.1612	0.0039	600.9281
940	236.1367	12,961.3623	116.1232	0.0359	600.9422	996	236.0183	12,898.8506	116.2109	0.0104	598.0439
941	235.7633	12,894.4990	116.2070	-0.0446	597.8422	997	236.4350	12,925.6758	116.1532	0.1002	599.2876
942	236.2033	12,947.7686	116.1059	0.0502	600.3120	998	236.0667	12,955.5635	116.0315	0.0208	600.6734
943	235.6533	12,858.0879	116.2932	-0.0684	596.1540	999	235.8267	12,968.8330	116.1320	-0.0308	601.2886
944	235.9683	12,982.0977	116.2423	-0.0004	601.9036	1,000	235.7550	12,914.7803	116.2426	-0.0463	598.7825
945	236.0033	12,985.8711	116.0300	0.0072	602.0786						
946	235.7383	12,864.7598	116.1200	-0.0500	596.4633						
947	236.1350	12,931.7012	116.1700	0.0355	599.5670						
948	235.7683	12,844.4033	116.4000	-0.0436	595.5195						
949	236.1950	12,944.7236	116.1300	0.0484	600.1708						
950	235.9687	12,859.6797	116.1200	-0.0007	596.2278						
951	235.7783	12,920.8350	116.0900	-0.0413	599.0632						
952	236.1000	12,878.4375	116.2300	0.0281	597.0975						
TOTAL	235.9758	12,918.4658	116.1528	0.0013	598.9534						
Max	236.7883	13,065.1729	116.4480	0.1760	605.7553						
Min	235.2200	12,702.5879	115.8885	-0.1623	588.9444						

ตาราง 30 เปรียบเทียบค่าคะแนนเฉลี่ยและความแปรปรวนของค่าความยาวของลำตัวโดยวิธีการสุ่มแบบเฉพาะวิภาคซึ่งอิงพหุคูณกับวิธีการสุ่มแบบง่าย

กรณีศึกษาของปลาก่อนและภายหลังการปล่อยความยาวของลำตัว จำนวนปลาคู่ละ 10 ตัว จำนวนปลาทั้งหมด 60 ตัว

MMS		SRS				MMS		SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	variance	Mean	variance	t-value	F-value
1	235.9817	12,949.5527	234.3467	12,952.4004	0.2488	1.0002	57	235.9533	13,011.9131	235.4883	12,850.0146	0.0708	1.0126
2	235.9267	13,025.8057	240.1900	13,538.3408	-0.6407	1.0393	58	236.3683	12,975.9209	237.2483	13,196.9834	-0.1332	1.0170
3	235.9550	12,905.9502	231.7250	12,426.1943	0.6510	1.0386	59	235.6867	12,909.6064	238.7183	13,144.0977	-0.4601	1.0182
4	235.9000	12,892.4902	235.4483	13,484.8232	0.0681	1.0459	60	235.9650	12,891.6768	234.4133	13,145.2979	0.2355	1.0197
5	235.9817	12,984.7666	238.0933	12,650.2217	-0.4157	1.0346	61	236.4583	12,891.1729	233.3300	13,214.0547	0.4743	1.0250
6	236.1367	12,859.5020	238.9833	13,005.8027	-0.4336	1.0114	62	236.1050	12,913.0400	231.0267	12,486.1299	0.7805	1.0342
7	235.8317	12,856.4795	242.2850	12,651.7588	-0.9897	1.0162	63	235.9200	12,973.1592	242.0183	12,951.5859	-0.9277	1.0017
8	236.1317	13,012.0283	235.9967	12,172.5674	0.0208	1.0690	64	235.7233	12,875.6299	233.4000	12,791.9736	0.3552	1.0065
9	235.7917	12,841.2891	233.3750	13,017.0645	0.3681	1.0137	65	235.6267	12,912.1768	232.8700	13,426.6611	0.4161	1.0398
10	236.1000	12,838.3154	241.4167	13,187.9531	-0.8073	1.0272	66	236.5217	12,912.5283	236.6600	12,774.4951	-0.0211	1.0108
11	236.0433	12,872.4854	237.2517	13,016.5811	-0.1840	1.0112	67	236.1983	12,961.0576	232.6050	13,212.2881	0.5441	1.0194
12	236.1100	12,914.5039	236.9917	12,987.8760	-0.1342	1.0057	68	236.1000	12,832.2754	240.3700	13,246.3135	-0.6477	1.0323
13	235.7600	12,937.7900	234.7017	12,920.8291	0.1612	1.0013	69	235.7783	12,965.3135	238.8217	13,032.9932	-0.4623	1.0052
14	236.4900	12,877.5742	232.2900	12,783.1543	0.6422	1.0074	70	236.0633	12,926.1299	229.2400	12,263.8760	1.0531	1.0540
15	235.7983	12,895.7041	234.7100	12,596.1631	0.1670	1.0238	71	235.8367	12,926.6309	238.1233	13,162.6826	-0.3468	1.0183
16	236.1767	12,944.8301	231.4083	13,430.7607	0.7192	1.0375	72	236.1467	12,975.9092	241.1450	12,492.2432	-0.7672	1.0387
17	236.0467	13,016.6821	234.2617	13,368.4727	0.2692	1.0270	73	236.1400	12,884.3174	241.4183	12,002.4590	-0.8196	1.0735
18	236.0667	12,947.7822	238.8650	13,476.9004	-0.4217	1.0409	74	236.0367	12,862.0527	237.7267	12,998.5293	-0.2574	1.0106
19	235.8517	12,967.5527	237.1533	13,084.5039	-0.1975	1.0090	75	235.7767	12,939.0879	236.6583	13,005.3223	-0.1341	1.0051
20	236.0750	12,813.8428	243.5483	12,891.0547	-1.1418	1.0060	76	235.8867	12,921.7129	237.3483	13,879.7080	-0.2187	1.0741
21	235.8060	12,976.5068	235.3867	13,080.7588	0.0635	1.0080	77	236.2417	12,913.2529	235.0250	12,825.9795	0.1858	1.0068
22	236.0450	12,896.1455	231.8783	12,216.3525	0.6441	1.0556	78	235.6883	12,945.4873	227.5883	12,873.4199	1.2348	1.0056
23	235.9400	12,871.7275	233.1767	13,198.7930	0.4192	1.0254	79	236.1200	12,875.7949	235.1867	13,188.8604	0.1416	1.0243
24	236.0400	12,961.2891	246.1300	13,194.4873	-1.5282	1.0180	80	235.9267	12,920.6328	240.9450	12,722.0771	-0.7676	1.0156
25	235.8333	12,804.8750	235.1367	13,296.0146	0.1054	1.0303	81	236.1650	12,849.5137	230.0317	11,931.7820	0.9544	1.0769
26	236.3183	12,979.3838	235.7583	13,365.6240	0.0845	1.0298	82	235.7817	12,852.6250	236.1533	13,139.3018	-0.0565	1.0223
27	236.1850	12,893.6406	234.8583	13,324.1631	0.2007	1.0334	83	236.0017	12,963.6553	235.1933	13,113.2646	0.1226	1.0115
28	235.4250	12,871.2764	236.4117	12,463.8105	-0.1518	1.0327	84	235.8200	12,921.2949	234.6417	12,827.5811	0.1799	1.0073
29	235.8550	12,955.0645	237.2117	13,100.1719	-0.2059	1.0112	85	236.1817	12,998.3887	236.6067	13,339.2236	-0.0641	1.0262
30	236.1117	12,926.7012	236.0250	12,877.5752	0.0132	1.0038	86	236.5467	12,910.2949	237.1000	12,908.4238	-0.0844	1.0001
31	236.1967	12,849.4951	224.3550	13,149.7100	1.7955	1.0155	87	235.9083	12,986.3457	230.6383	12,507.3984	0.8085	1.0383
32	235.9450	12,956.9854	235.1150	13,000.4443	0.1262	1.0034	88	235.9483	12,917.3037	242.1833	13,474.8379	-0.9401	1.0432
33	235.6717	12,957.6230	237.4767	13,113.8766	-0.2738	1.0121	89	236.0617	12,900.6748	232.2200	12,641.1602	0.5888	1.0205
34	235.9350	12,940.4502	234.5100	13,502.3770	0.2147	1.0434	90	235.9133	12,920.0898	232.5050	13,392.3252	0.5147	1.0366
35	235.8683	12,978.1543	234.8267	12,806.8594	0.1436	1.0134	91	235.5750	12,888.5195	238.9867	12,961.3291	-0.5198	1.0056
36	236.2400	12,790.0703	232.8700	12,671.3154	0.5173	1.0094	92	236.1433	12,964.2227	237.1717	12,232.3281	-0.1587	1.0598
37	235.5700	12,874.2207	239.5033	13,230.8916	-0.5963	1.0277	93	235.6617	12,870.7549	236.3083	12,235.5713	-0.1000	1.0519
38	235.6133	12,895.4512	233.6333	12,979.1777	0.3015	1.0065	94	235.7867	13,015.3945	233.5783	12,654.3330	0.3376	1.0285
39	236.6050	12,923.6328	237.6750	14,012.1611	-0.3089	1.0842	95	236.1067	12,723.3350	235.1950	12,069.8086	0.1418	1.0541
40	235.8617	12,927.8359	234.0600	13,739.5762	0.2702	1.0628	96	235.7600	12,940.1309	234.9150	12,913.6055	0.1287	1.0021
41	235.9933	12,976.5205	240.2283	12,803.1260	-0.6461	1.0135	97	235.9833	12,960.7617	235.6583	12,848.7383	0.0496	1.0087
42	235.7150	12,937.1211	236.9850	12,488.7373	-0.1951	1.0359	98	236.0317	12,887.2373	231.9083	12,581.1465	0.6329	1.0243
43	235.9600	12,958.0430	238.5450	12,778.4805	-0.3947	1.0141	99	236.3217	12,985.4063	241.3417	12,865.1621	-0.7648	1.0093
44	236.3400	12,898.4736	239.8417	12,547.3408	-0.5377	1.0280	100	235.8550	12,927.6006	230.9217	13,187.4160	0.7478	1.0201
45	236.1617	12,981.1631	226.7183	12,730.2344	1.4426	1.0197	101	236.1367	12,930.1670	233.2633	13,235.0420	0.4351	1.0236
46	235.8783	12,989.8545	233.1133	13,144.3809	0.4190	1.0119	102	236.2217	12,965.3311	233.6083	12,619.9531	0.4002	1.0274
47	235.9533	12,953.2500	232.7900	13,102.3936	0.4800	1.0115	103	235.9783	13,021.1895	237.3733	13,698.9727	-0.2090	1.0521
48	235.7233	12,882.1104	236.7983	12,926.6338	-0.1639	1.0035	104	236.1483	12,971.1729	235.0000	13,112.6006	0.1742	1.0109
49	236.0900	12,864.1436	238.5000	12,772.4912	-0.3687	1.0072	105	235.7017	12,837.0615	239.4167	12,812.2002	-0.5682	1.0019
50	235.6600	12,857.6709	234.0033	12,608.1738	0.2543	1.0198	106	236.2767	12,883.6895	233.4200	12,964.8652	0.4352	1.0063
51	235.9183	12,835.6729	239.8950	12,630.3525	-0.6104	1.0163	107	236.2783	12,972.8555	235.1400	13,822.0107	0.1703	1.0655
52	235.8217	12,917.9492	239.5417	13,020.4844	-0.5658	1.0079	108	236.1750	12,933.0244	232.2683	12,448.6963	0.6007	1.0389
53	236.1183	12,996.8277	236.7683	13,389.8789	-0.0980	1.0302	109	235.9550	12,922.4561	236.8000	12,341.6631	-0.1302	1.0471
54	235.9017	12,949.6230	235.5900	13,038.8770	0.0474	1.0069	110	235.7917	12,898.5674	237.0650	12,813.5283	-0.1945	1.0066
55	236.2667	12,950.2900	236.4950	12,913.2100	-0.0348	1.0029	111	235.6033	12,926.4346	239.4717	13,389.6172	-0.5841	1.0358
56	236.0833	12,968.0420	238.1617	13,277.4365	-0.3142	1.0239	112	235.5567	12,845.2676	231.2400	13,555.9492	0.6507	1.0553

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่สุ่มโดยเทคนิคการสุ่มแบบแบ่งครึ่งเชิงพหุคูณกับเทคนิคการสุ่มแบบง่าย
กรณีข้อมูลมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

MMS						SRS						MMS						SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	variance	Mean	variance	t-value	F-value	No.	Mean	Variance	Mean	variance	t-value	F-value			
113	236.2417	12,975.0684	239.7717	13,289.1836	-0.5335	1.0242	169	236.0950	12,911.7373	240.0783	13,745.0820	-0.5976	1.0645	170	236.0983	12,944.2373	234.7217	12,555.6455	0.2112	1.0309			
114	236.2150	13,009.9551	233.2283	12,543.1494	0.4577	1.0372	170	236.0983	12,944.2373	234.7217	12,555.6455	0.2112	1.0309	171	236.0917	12,898.7988	239.0850	12,529.5449	-0.4598	1.0295			
115	235.7550	12,826.9033	239.6533	12,618.1572	-0.5986	1.0165	171	236.0917	12,898.7988	239.0850	12,529.5449	-0.4598	1.0295	172	235.7517	12,951.1221	236.3767	13,339.0762	-0.0944	1.0300			
116	235.9683	12,802.7764	234.2483	13,299.4639	0.2608	1.0388	172	235.7517	12,951.1221	236.3767	13,339.0762	-0.0944	1.0300	173	236.3217	13,028.0313	241.0750	13,352.3984	-0.7169	1.0249			
117	235.4617	13,028.7520	235.9200	12,340.3340	-0.0705	1.0558	173	236.3217	13,028.0313	241.0750	13,352.3984	-0.7169	1.0249	174	236.0017	12,940.1035	235.0283	13,200.7568	0.1475	1.0201			
118	236.1317	12,905.3301	233.3517	13,153.7959	0.4218	1.0193	174	236.0017	12,940.1035	235.0283	13,200.7568	0.1475	1.0201	175	236.0533	13,023.1709	239.7367	13,901.4531	-0.5498	1.0674			
119	235.6517	12,918.9092	238.7550	13,036.3105	-0.4718	1.0091	175	236.0533	13,023.1709	239.7367	13,901.4531	-0.5498	1.0674	176	235.9367	12,948.0469	235.3883	13,093.9160	0.0832	1.0111			
120	236.2567	12,906.4082	234.3200	13,133.6934	0.2940	1.0176	176	235.9367	12,948.0469	235.3883	13,093.9160	0.0832	1.0111	177	235.9083	12,995.7803	240.5550	13,194.5293	-0.7033	1.0153			
121	235.9267	12,925.1709	237.5733	12,765.7813	-0.2516	1.0125	177	235.9083	12,995.7803	240.5550	13,194.5293	-0.7033	1.0153	178	235.7033	12,988.0664	230.3267	13,137.0947	0.8148	1.0115			
122	236.0433	12,910.8789	237.5900	12,651.3379	-0.2370	1.0205	178	235.7033	12,988.0664	230.3267	13,137.0947	0.8148	1.0115	179	236.0933	12,917.3525	236.0867	12,967.5381	0.0010	1.0039			
123	236.0200	12,901.9258	237.6300	12,987.1416	-0.2451	1.0068	179	236.0933	12,917.3525	236.0867	12,967.5381	0.0010	1.0039	180	235.4800	12,889.9053	235.1967	12,784.7764	0.0433	1.0082			
124	236.1233	12,856.8359	239.1100	13,284.0410	-0.4525	1.0332	180	235.4800	12,889.9053	235.1967	12,784.7764	0.0433	1.0082	181	235.9033	12,893.7314	240.9467	13,709.3125	-0.7574	1.0633			
125	236.1950	12,954.2178	236.2733	13,285.8818	-0.0118	1.0256	181	235.9033	12,893.7314	240.9467	13,709.3125	-0.7574	1.0633	182	236.0150	12,827.6973	233.7133	12,755.1836	0.3525	1.0057			
126	236.1283	12,890.1641	236.2717	13,194.5303	-0.0217	1.0204	182	236.0150	12,827.6973	233.7133	12,755.1836	0.3525	1.0057	183	235.6500	12,854.6318	231.3367	13,427.6924	0.6517	1.0446			
127	236.1833	12,927.5449	231.1333	12,750.6802	0.7719	1.0139	183	235.6500	12,854.6318	231.3367	13,427.6924	0.6517	1.0446	184	236.1733	12,840.0254	231.8350	12,890.1631	0.6625	1.0039			
128	235.7900	12,859.4521	234.8783	12,098.3594	0.1414	1.0629	184	236.1733	12,840.0254	231.8350	12,890.1631	0.6625	1.0039	185	236.1750	12,940.3613	236.0217	12,849.0215	0.0234	1.0071			
129	235.7983	12,875.0361	236.8733	13,192.8057	-0.1631	1.0247	185	236.1750	12,940.3613	236.0217	12,849.0215	0.0234	1.0071	186	235.6050	12,858.6133	237.6900	13,549.1055	-0.3143	1.0537			
130	235.5950	12,946.7686	234.5883	12,431.8135	0.1548	1.0414	186	235.6050	12,858.6133	237.6900	13,549.1055	-0.3143	1.0537	187	235.8483	12,887.2578	237.4083	12,039.1553	-0.2420	1.0704			
131	236.3033	12,927.0557	238.5467	13,557.5439	-0.3377	1.0488	187	235.8483	12,887.2578	237.4083	12,039.1553	-0.2420	1.0704	188	235.8150	12,998.9854	236.9183	12,933.4658	-0.1678	1.0051			
132	235.9300	12,773.0586	236.3183	12,910.1426	-0.0594	1.0107	188	235.8150	12,998.9854	236.9183	12,933.4658	-0.1678	1.0051	189	235.9917	12,952.6436	239.5650	12,746.3418	-0.5460	1.0162			
133	236.1533	12,919.2373	237.3633	12,705.4902	-0.1852	1.0168	189	235.9917	12,952.6436	239.5650	12,746.3418	-0.5460	1.0162	190	235.8200	12,971.1504	234.2283	13,019.6006	0.2418	1.0037			
134	236.0367	13,008.0186	239.8883	13,275.0059	-0.5820	1.0205	190	235.8200	12,971.1504	234.2283	13,019.6006	0.2418	1.0037	191	236.1783	12,906.4463	234.9100	12,268.7930	0.1958	1.0520			
135	236.0200	12,935.3711	236.7400	13,979.0439	-0.1087	1.0343	191	236.1783	12,906.4463	234.9100	12,268.7930	0.1958	1.0520	192	236.6550	12,903.7764	234.7233	13,361.6328	0.2920	1.0355			
136	236.1900	12,908.9287	237.0817	12,540.9111	-0.1384	1.0293	192	236.6550	12,903.7764	234.7233	13,361.6328	0.2920	1.0355	193	235.9117	12,912.9160	237.6133	12,164.1904	-0.2632	1.0616			
137	235.9800	12,969.2861	235.8983	13,272.4639	0.0124	1.0234	193	235.9117	12,912.9160	237.6133	12,164.1904	-0.2632	1.0616	194	235.7500	12,962.8584	234.1783	13,008.6025	0.2389	1.0035			
138	235.2917	12,898.7402	238.9500	12,785.5537	-0.5591	1.0089	194	235.7500	12,962.8584	234.1783	13,008.6025	0.2389	1.0035	195	236.1817	12,939.6523	236.1267	12,609.7598	0.0084	1.0262			
139	235.8017	12,929.9658	237.2617	13,185.9180	-0.2213	1.0198	195	236.1817	12,939.6523	236.1267	12,609.7598	0.0084	1.0262	196	235.6633	12,919.5674	237.8567	13,395.9863	-0.3312	1.0369			
140	235.6617	12,943.2852	238.6250	13,379.4287	-0.4474	1.0337	196	235.6633	12,919.5674	237.8567	13,395.9863	-0.3312	1.0369	197	236.1433	12,929.3301	238.3017	12,456.6504	-0.3318	1.0379			
141	235.8083	12,850.8457	232.5150	12,199.3066	0.5097	1.0534	197	236.1433	12,929.3301	238.3017	12,456.6504	-0.3318	1.0379	198	235.7817	12,826.0449	235.2167	12,739.2246	0.0886	1.0068			
142	235.8233	12,829.7939	238.8767	13,005.9277	-0.4663	1.0137	198	235.7817	12,826.0449	235.2167	12,739.2246	0.0886	1.0068	199	235.6600	12,937.1279	241.3133	12,631.7480	-0.8660	1.0242			
143	235.9017	12,873.7285	232.8333	13,551.9121	0.4623	1.0527	199	235.6600	12,937.1279	241.3133	12,631.7480	-0.8660	1.0242	200	235.8467	12,884.7061	241.8733	13,324.9590	-0.9118	1.0342			
144	235.6917	12,943.9248	233.6000	12,924.4473	0.3186	1.0015	200	235.8467	12,884.7061	241.8733	13,324.9590	-0.9118	1.0342	201	236.3183	12,905.7783	237.1100	12,821.1963	-0.1209	1.0066			
145	236.0183	12,906.7900	235.6733	12,204.8418	0.0533	1.0575	201	236.3183	12,905.7783	237.1100	12,821.1963	-0.1209	1.0066	202	235.6833	12,997.1816	235.8117	13,139.7041	-0.0194	1.0110			
146	236.1883	12,920.8535	236.2150	13,058.1240	-0.0041	1.0106	202	235.6833	12,997.1816	235.8117	13,139.7041	-0.0194	1.0110	203	235.8783	12,902.6650	232.1483	12,923.3906	0.5685	1.0016			
147	235.8150	12,975.7959	238.7950	12,403.0195	-0.4582	1.0462	203	235.8783	12,902.6650	232.1483	12,923.3906	0.5685	1.0016	204	236.1583	12,959.5684	237.9383	12,456.1094	-0.2735	1.0404			
148	236.4200	13,016.7354	234.5950	13,240.9209	0.2759	1.0172	204	236.1583	12,959.5684	237.9383	12,456.1094	-0.2735	1.0404	205	235.8317	12,911.5625	235.9067	13,368.3516	-0.0113	1.0354			
149	235.3467	12,990.5352	235.5650	13,084.7783	-0.0331	1.0073	205	235.8317	12,911.5625	235.9067	13,368.3516	-0.0113	1.0354	206	235.7750	12,926.4756	237.4067	12,547.1670	-0.2504	1.0302			
150	235.7900	12,919.6055	238.2883	13,395.2285	-0.3863	1.0368	206	235.7750	12,926.4756	237.4067	12,547.1670	-0.2504	1.0302	207	236.1500	12,941.3574	241.2250	12,958.5234	-0.7724	1.0013			
151	235.8217	12,916.0527	234.5617	13,777.6738	0.1889	1.0667	207	236.1500	12,941.3574	241.2250	12,958.5234	-0.7724	1.0013	208	236.1617	12,948.4053	238.7350	12,386.5908	-0.3960	1.0454			
152	235.9200	12,897.2266	237.4133	13,130.3193	-0.2267	1.0181	208	236.1617	12,948.4053	238.7350													

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่สุ่มโดยเทคนิคการสุ่มแบบเมตริกซ์เชิงพหุคูณแบบเทคนิคกับวิธีการสุ่มแบบง่าย

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

MMS						SRS						MMS						SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value			
225	236.2533	12,857.8203	241.2650	13,247.1514	-0.7598	1.0303	281	236.0750	12,945.9766	239.1500	12,989.0723	-0.4677	1.0033	281	236.0750	12,945.9766	239.1500	12,989.0723	-0.4677	1.0033			
226	235.8067	12,945.6250	231.1600	13,051.6006	0.7059	1.0082	282	236.2100	12,946.7168	232.0767	13,471.4268	0.6229	1.0405	282	236.2100	12,946.7168	232.0767	13,471.4268	0.6229	1.0405			
227	236.0617	13,000.6641	233.9583	12,779.2842	0.3209	1.0173	283	236.0950	12,867.8438	238.9900	13,238.0430	-0.4389	1.0288	283	236.0950	12,867.8438	238.9900	13,238.0430	-0.4389	1.0288			
228	235.7600	12,868.8242	229.7633	13,535.9502	0.9039	1.0518	284	235.7700	12,880.7813	231.1267	12,200.1240	0.7182	1.0558	284	235.7700	12,880.7813	231.1267	12,200.1240	0.7182	1.0558			
229	236.3100	12,944.2959	230.8933	13,287.1006	0.8192	1.0265	285	236.0067	12,940.5791	239.1500	13,522.4482	-0.4733	1.0450	285	236.0067	12,940.5791	239.1500	13,522.4482	-0.4733	1.0450			
230	236.6833	12,948.1172	237.3717	13,158.4756	-0.1044	1.0162	286	236.1467	12,923.5684	235.6783	13,018.4941	0.0712	1.0073	286	236.1467	12,923.5684	235.6783	13,018.4941	0.0712	1.0073			
231	236.3750	12,869.4453	233.2167	12,706.2500	0.4837	1.0128	287	236.0667	12,897.6172	231.7233	13,015.5098	0.6609	1.0091	287	236.0667	12,897.6172	231.7233	13,015.5098	0.6609	1.0091			
232	235.4700	12,949.4326	236.1150	12,096.2334	-0.0998	1.0705	288	236.3050	12,928.8926	237.5983	12,525.4277	-0.1986	1.0322	288	236.3050	12,928.8926	237.5983	12,525.4277	-0.1986	1.0322			
233	236.0800	12,948.2705	238.4933	13,085.3721	-0.3664	1.0106	289	236.1567	12,981.2197	237.9383	12,951.6621	-0.2710	1.0023	289	236.1567	12,981.2197	237.9383	12,951.6621	-0.2710	1.0023			
234	235.7750	12,862.8750	229.9050	12,409.8008	0.9045	1.0365	290	235.8350	12,872.9727	233.1083	12,878.5088	0.4162	1.0004	290	235.8350	12,872.9727	233.1083	12,878.5088	0.4162	1.0004			
235	236.1367	12,920.5156	239.1867	12,354.4492	-0.4699	1.0458	291	236.1883	12,890.9023	229.8117	12,771.4902	0.9750	1.0093	291	236.1883	12,890.9023	229.8117	12,771.4902	0.9750	1.0093			
236	235.9633	12,798.5986	237.0183	12,555.4219	-0.1623	1.0194	292	236.1000	12,919.0605	236.1767	12,665.9053	-0.0117	1.0200	292	236.1000	12,919.0605	236.1767	12,665.9053	-0.0117	1.0200			
237	236.0567	12,991.7734	230.8233	12,637.1240	0.8007	1.0281	293	235.9433	12,931.7432	237.1850	13,222.9609	-0.1881	1.0225	293	235.9433	12,931.7432	237.1850	13,222.9609	-0.1881	1.0225			
238	235.9217	12,894.1309	239.7983	12,997.1948	-0.5901	1.0080	294	236.1850	12,971.8926	232.9433	13,210.1133	0.4907	1.0184	294	236.1850	12,971.8926	232.9433	13,210.1133	0.4907	1.0184			
239	236.3400	12,841.9281	239.0167	13,345.9297	-0.4052	1.0393	295	236.3817	12,920.9678	235.0433	12,873.7813	0.2041	1.0037	295	236.3817	12,920.9678	235.0433	12,873.7813	0.2041	1.0037			
240	236.0833	12,988.5869	236.3483	13,222.1621	-0.0401	1.0180	296	236.2117	12,971.0410	236.2150	12,693.6563	-0.0005	1.0219	296	236.2117	12,971.0410	236.2150	12,693.6563	-0.0005	1.0219			
241	235.6750	12,873.8242	237.5933	13,111.5137	-0.3067	1.0185	297	236.3000	12,872.2686	245.0400	12,420.8398	-1.3461	1.0363	297	236.3000	12,872.2686	245.0400	12,420.8398	-1.3461	1.0363			
242	236.3483	12,983.4766	239.3833	12,939.7129	-0.4617	1.0034	298	235.8967	12,848.4785	239.5017	12,820.8535	-0.5512	1.0022	298	235.8967	12,848.4785	239.5017	12,820.8535	-0.5512	1.0022			
243	235.6900	12,942.5791	231.9450	12,756.9316	0.5722	1.0146	299	235.9050	13,001.1094	236.4267	12,894.1553	-0.0794	1.0083	299	235.9050	13,001.1094	236.4267	12,894.1553	-0.0794	1.0083			
244	235.5717	12,939.5049	236.3617	12,828.3301	-0.1205	1.0087	300	235.8867	12,882.5645	232.8167	13,003.5527	0.4674	1.0094	300	235.8867	12,882.5645	232.8167	13,003.5527	0.4674	1.0094			
245	235.7750	12,893.3193	238.1533	12,591.3691	-0.3649	1.0240	301	236.1850	12,935.6328	234.5717	12,472.2266	0.2479	1.0372	301	236.1850	12,935.6328	234.5717	12,472.2266	0.2479	1.0372			
246	235.5350	12,971.4756	233.2767	12,708.0098	0.3452	1.0207	302	235.7917	12,904.4805	233.3067	12,746.5439	0.3801	1.0124	302	235.7917	12,904.4805	233.3067	12,746.5439	0.3801	1.0124			
247	236.3183	12,955.5950	234.5333	13,229.4648	0.2702	1.0211	303	235.5267	12,923.0371	237.5417	13,166.3008	-0.3056	1.0188	303	235.5267	12,923.0371	237.5417	13,166.3008	-0.3056	1.0188			
248	235.8033	12,970.1758	231.4017	12,750.4395	0.6723	1.0172	304	236.0650	12,872.1709	237.8133	12,733.3008	-0.2676	1.0109	304	236.0650	12,872.1709	237.8133	12,733.3008	-0.2676	1.0109			
249	236.1200	12,853.2207	229.0717	12,526.8291	1.0837	1.0261	305	235.7883	12,993.4854	237.9367	13,214.5000	-0.3251	1.0170	305	235.7883	12,993.4854	237.9367	13,214.5000	-0.3251	1.0170			
250	236.4317	12,835.4697	235.7800	13,003.3770	0.0993	1.0131	306	235.2533	12,879.0801	237.2700	13,822.6182	-0.1524	1.0733	306	235.2533	12,879.0801	237.2700	13,822.6182	-0.1524	1.0733			
251	235.8833	12,880.8379	233.4900	13,227.1250	0.3628	1.0269	307	235.9250	12,921.5342	242.1067	13,580.4893	-0.9301	1.0510	307	235.9250	12,921.5342	242.1067	13,580.4893	-0.9301	1.0510			
252	235.8533	12,971.8662	234.8000	12,620.2871	0.1613	1.0279	308	236.1450	12,944.8252	236.3300	13,386.1543	-0.0279	1.0341	308	236.1450	12,944.8252	236.3300	13,386.1543	-0.0279	1.0341			
253	235.9817	12,906.5479	236.8283	13,708.7686	-0.1271	1.0622	309	236.0450	12,858.5625	226.1767	12,872.4834	1.5069	1.0011	309	236.0450	12,858.5625	226.1767	12,872.4834	1.5069	1.0011			
254	235.5133	12,967.0439	231.7417	13,225.6289	0.5719	1.0279	310	236.2267	12,856.2852	233.3350	13,749.5469	0.4342	1.0695	310	236.2267	12,856.2852	233.3350	13,749.5469	0.4342	1.0695			
255	236.2583	12,818.6211	238.6300	13,534.4941	-0.3579	1.0558	311	236.7783	12,842.7227	235.6400	12,552.3447	0.0213	1.0231	311	236.7783	12,842.7227	235.6400	12,552.3447	0.0213	1.0231			
256	235.9783	12,924.2939	235.6650	12,841.8711	0.0478	1.0064	312	235.5900	12,994.7588	229.5550	12,443.9951	0.9268	1.0443	312	235.5900	12,994.7588	229.5550	12,443.9951	0.9268	1.0443			
257	236.2417	12,944.0078	237.8783	13,934.5557	-0.2473	1.0302	313	235.9400	12,935.2637	239.5217	13,612.9469	-0.5384	1.0524	313	235.9400	12,935.2637	239.5217	13,612.9469	-0.5384	1.0524			
258	236.3133	12,967.8340	236.1550	12,797.3652	0.0242	1.0133	314	236.0587	12,850.4160	237.4067	13,124.6826	-0.2052	1.0213	314	236.0587	12,850.4160	237.4067	13,124.6826	-0.2052	1.0213			
259	236.0147	12,930.6631	236.8567	13,090.0586	-0.0934	1.0123	315	236.3350	12,953.1924	238.7867	12,857.8104	-0.3738	1.0074	315	236.3350	12,953.1924	238.7867	12,857.8104	-0.3738	1.0074			
260	236.0267	12,896.7578	240.2367	12,507.2695	-0.6470	1.0311	316	235.9767	12,867.9551	230.9317	13,353.0000	0.7632	1.0377	316	235.9767	12,867.9551	230.9317	13,353.0000	0.7632	1.0377			
261	235.6983	12,946.7539	233.6950	13,716.9219	0.3005	1.0595	317	235.9017	12,998.2900	240.2733	13,063.6377	-0.6633	1.0050	317	235.9017	12,998.2900	240.2733	13,063.6377	-0.6633	1.0050			
262	236.0500	12,919.2109	241.4483	13,439.0938	-0.8145	1.0402	318	236.0400	12,890.6230	239.2017	12,648.5303	-0.4846	1.0191	318	236.0400	12,890.6230	239.2017	12,648.5303	-0.4846	1.0191			
263	236.1800	12,894.4932	236.7167	13,061.1084	-0.0816	1.0137	319	235.9717	12,923.1572	234.7367	13,021.5332	0.1878	1.0076	319	235.9717	12,923.1572	234.7367	13,021.5332	0.1878	1.0076			
264	235.8900	12,855.7666	243.6350	12,808.9082	-1.1842	1.0037	320	235.9950	12,919.2588	235.8083	13,205.2256	0.0283											

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่สุ่มโดยเทคนิคการสุ่มแบบมัลติกริฟฟิงกับเทคนิคการสุ่มแบบง่าย

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

MMS							SRS						
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value
337	235.9883	12.938.4541	232.3350	12.273.0225	0.5636	1.0542	393	236.1100	12.886.0586	234.3783	12.777.3994	0.2648	1.0085
338	236.1767	12.933.9531	239.0267	13.739.7988	-0.4274	1.0623	394	236.0183	12.869.8760	237.4600	12.673.1641	-0.2210	1.0155
339	235.8483	12.989.3555	232.8233	12.441.4043	0.4646	1.0440	395	235.9550	12.832.2295	238.1183	12.297.8623	-0.3343	1.0435
340	235.9950	12.863.9775	238.3433	12.694.9336	-0.3598	1.0133	396	236.1383	12.915.8672	235.4567	12.394.2754	0.1050	1.0421
341	236.2483	12.877.0518	229.9067	13.043.0029	0.9649	1.0129	397	236.3183	12.898.8203	238.3167	12.420.5469	-0.3076	1.0385
342	236.0850	13.030.2988	242.0567	13.226.7148	-0.9027	1.0151	398	236.1900	12.923.0303	232.3000	13.342.2539	0.5879	1.0324
343	235.5800	12.961.0313	230.6950	13.014.4746	0.7424	1.0041	399	235.6567	12.906.2363	235.5417	12.876.9854	0.0175	1.0023
344	235.4650	12.864.4336	237.6433	12.823.9131	-0.3329	1.0032	400	236.1117	12.933.4072	242.0533	12.386.5146	-0.9146	1.0442
345	236.4950	12.846.4629	229.7783	12.197.2266	1.0396	1.0532	401	235.9633	12.952.9590	238.1583	13.179.4072	-0.3326	1.0175
346	236.0717	12.919.7969	236.5767	12.689.0293	-0.0773	1.0198	402	236.1917	12.925.0850	233.1517	12.735.4990	0.4649	1.0149
347	236.2167	12.828.1035	233.2117	13.161.7148	0.4566	1.0260	403	235.7317	12.967.5908	238.0733	13.630.3447	-0.3517	1.0511
348	235.9167	12.964.0498	238.3950	12.687.1357	-0.3792	1.0234	404	235.7633	12.938.3535	240.2617	12.617.3867	-0.6893	1.0254
349	236.0233	12.922.5879	238.3917	12.785.5557	-0.3618	1.0107	405	235.6083	12.833.8633	238.1850	12.267.1172	-0.3984	1.0462
350	236.0567	12.915.4463	236.5233	13.387.1113	-0.0705	1.0365	406	235.5100	12.939.2266	240.3917	12.961.3223	-0.7430	1.0017
351	235.9050	12.888.2100	237.2733	12.029.3877	-0.2123	1.0714	407	236.1033	12.985.0195	235.4133	12.841.9121	0.1052	1.0111
352	236.0517	12.862.2578	240.6300	12.362.6074	-0.7061	1.0404	408	236.0650	12.913.3408	236.7567	13.169.0156	-0.1049	1.0198
353	236.1033	12.924.6660	231.8200	12.738.3311	0.6549	1.0146	409	235.8800	12.857.5537	240.6367	12.539.2871	-0.7311	1.0254
354	235.6733	12.979.3408	233.4167	13.181.8428	0.3418	1.0156	410	236.3550	12.879.8643	235.7233	12.587.5029	0.0970	1.0232
355	236.0517	12.999.7012	232.7050	13.128.0498	0.5072	1.0099	411	235.5817	12.864.1455	235.1783	13.701.8750	0.0606	1.0651
356	236.2250	12.918.1826	239.6050	13.389.6270	-0.5104	1.0365	412	235.7500	12.865.6318	237.1000	12.899.0947	-0.2060	1.0026
357	235.9000	12.913.5107	228.7417	12.609.8330	1.0975	1.0241	413	236.2667	12.871.2041	234.6267	13.183.6738	0.2489	1.0243
358	235.7767	12.867.0977	235.9850	12.685.7490	-0.0319	1.0143	414	236.1783	12.993.1143	239.2900	13.201.6650	-0.4709	1.0161
359	236.1517	12.921.6758	238.4633	13.430.9639	-0.3488	1.0394	415	235.9717	12.973.5830	240.0367	12.936.9473	-0.6186	1.0028
360	235.8200	12.908.2969	234.0283	13.003.5986	0.2726	1.0074	416	235.7216	12.943.9395	232.0100	13.124.5879	0.5631	1.0140
361	236.2533	12.937.5244	235.7050	13.070.2363	0.0833	1.0103	417	236.0833	12.886.4883	240.2717	12.121.8893	-0.6487	1.0631
362	236.0017	12.891.4102	231.5667	13.115.1572	0.6736	1.0174	418	236.1617	12.927.7344	234.8000	12.704.2471	0.2083	1.0176
363	236.1117	12.937.2598	234.8133	12.655.6982	0.1988	1.0222	419	235.7000	12.943.8154	244.8650	12.690.0088	-1.4022	1.0200
364	235.9567	12.894.8135	236.3933	12.995.9521	-0.0665	1.0078	420	236.0017	12.973.2969	234.2967	12.396.4463	0.2622	1.0465
365	235.9467	12.862.4600	231.9000	12.298.1533	0.6237	1.0540	421	235.9533	12.918.7070	236.3433	12.864.0889	-0.0595	1.0042
366	235.8867	12.897.7725	230.0950	13.479.3232	0.8735	1.0451	422	235.7783	12.950.2959	232.4617	13.285.6826	0.5016	1.0259
367	235.8250	12.849.1514	236.8867	12.421.0625	0.0059	1.0345	423	235.7733	13.008.1318	235.7283	13.604.9414	0.0069	1.0459
368	235.8000	12.902.7949	238.2817	12.581.3965	-0.3808	1.0255	424	236.2934	12.939.7383	233.9550	12.009.8340	0.3626	1.0774
369	235.9250	12.916.0049	238.9267	13.304.5352	-0.4541	1.0301	425	235.9850	12.921.7676	235.4583	12.917.4688	0.0003	1.0003
370	236.1000	12.980.9033	238.5733	13.272.9932	-0.3739	1.0225	426	235.9950	12.934.6328	230.2750	12.250.2744	0.8829	1.0559
371	236.0533	12.940.6602	234.2250	13.224.2295	0.2769	1.0219	427	235.9633	12.906.8428	234.4117	13.106.4951	0.2357	1.0155
372	236.0517	12.983.9297	237.1533	12.493.2656	-0.1691	1.0393	428	236.4400	12.972.5391	240.6150	12.685.0371	-0.6384	1.0227
373	236.0083	12.871.4072	233.8900	13.248.7156	0.3211	1.0293	429	235.9267	12.902.3330	236.5817	12.492.2090	-0.4081	1.0328
374	236.2017	12.867.5557	235.0367	12.400.6064	0.1795	1.0377	430	235.7800	12.887.3926	240.0033	13.347.2520	-0.6386	1.0349
375	235.7500	12.945.2793	231.1533	12.527.0684	0.7055	1.0334	431	235.7067	12.934.1748	232.0383	13.158.4229	0.5563	1.0173
376	236.2067	12.844.2627	232.0900	12.739.7051	0.6304	1.0082	432	235.9367	12.923.5566	238.5900	12.691.6982	-0.4061	1.0183
377	236.3300	12.999.2959	235.4050	12.952.2197	0.1406	1.0036	433	236.3733	12.951.9365	232.6533	12.343.7705	0.5727	1.0493
378	235.6817	12.942.7920	234.3417	12.898.8145	0.2042	1.0034	434	235.8783	12.963.9893	237.9667	13.154.9307	-0.3165	1.0147
379	235.8817	12.905.1689	236.9067	13.685.4336	-0.1540	1.0589	435	236.0183	12.944.0010	235.5317	12.532.1143	0.0747	1.0329
380	235.7950	12.928.1924	232.0883	13.573.5352	0.5577	1.0499	436	236.0283	12.865.2764	237.6600	12.998.2148	-0.2485	1.0103
381	236.0650	12.921.4766	235.2667	12.972.5293	0.1215	1.0040	437	235.7483	12.810.3369	234.8283	12.792.1807	0.1408	1.0014
382	236.1500	12.932.0635	233.0933	13.184.8799	0.4633	1.0195	438	235.4100	12.901.9189	238.0600	12.732.9375	-0.4054	1.0133
383	235.6100	12.867.2744	241.4550	12.373.6260	-0.9012	1.0399	439	236.2083	12.903.0029	234.9350	12.643.0244	0.1951	1.0206
384	235.7500	12.905.8955	239.1800	13.267.9346	-0.5193	1.0281	440	236.2367	12.896.9600	234.4050	13.502.4004	0.2761	1.0459
385	235.3267	12.925.4707	230.5317	13.178.7764	0.7270	1.0196	441	236.2250	12.935.0088	234.6533	12.900.8682	0.2395	1.0026
386	235.4600	12.863.8574	232.8750	12.839.7275	0.3949	1.0019	442	235.7650	12.990.4541	237.2283	13.163.2637	-0.2216	1.0133
387	235.9883	12.931.4717	240.3567	12.575.7520	-0.6700	1.0283	443	236.1067	12.804.6426	237.3300	12.647.2363	-0.1878	1.0124
388	236.0383	12.887.7266	233.1433	12.370.0361	0.4462	1.0419	444	236.1917	12.962.1836	233.8167	11.854.1230	0.3693	1.0395
389	235.5017	12.886.2041	235.3567	13.226.9375	0.0220	1.0264	445	235.8017	12.923.0723	236.0483	13.385.5674	-0.0372	1.0358
390	236.1850	12.874.8887	238.0150	13.313.9902	-0.2770	1.0341	446	236.0517	12.908.3965	237.8617	12.220.1143	-0.2797	1.0563
391	235.6867	12.921.9863	239.1100	13.089.0498	-0.5199	1.0129	447	236.0550	12.928.0049	237.1317	12.755.4980	-0.1646	1.0185
392	235.9883	12.955.5762	237.4217	13.685.4346	-0.2151	1.0563	448	235.9233	12.842.3975	234.4333	13.054.7764	0.2268	1.0165

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่สุ่มโดยเทคนิคการสุ่มแบบเมตริกซ์เชิงพหุคูณกับเทคนิคการสุ่มแบบง่าย

กรณีข้อสอบมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน											
MMS						SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	variance	Mean	variance
449	235.9233	12,956.4756	234.4183	13,218.5693	0.2279	1.0202	505	236.2167	12,847.1074	234.4017	12,164.0586
450	236.1867	12,857.6318	234.7583	11,584.7061	0.2238	1.1099	506	236.3783	12,949.6768	239.1267	13,089.5527
451	236.0300	12,937.8281	240.9433	13,403.2021	-0.7415	1.0360	507	236.1417	13,027.2412	237.1700	12,994.5088
452	235.8200	12,927.1904	236.8200	13,362.8125	-0.1511	1.0337	508	236.0850	12,867.2988	237.6033	12,832.6035
453	236.3400	13,024.7637	235.7883	13,095.2275	0.0836	1.0054	509	236.1650	12,970.1055	236.3833	12,678.1504
454	235.6566	12,903.0264	232.8967	12,341.0645	0.4255	1.0455	510	236.5117	13,022.9277	237.8617	13,045.8037
455	235.7216	12,925.2012	235.4850	12,961.7246	0.0360	1.0028	511	236.0400	12,866.0459	234.5750	13,198.7139
456	235.9450	12,952.0576	241.4533	13,198.5322	-0.8348	1.0214	512	235.8567	12,953.0645	235.5267	13,194.1396
457	235.7817	12,866.1934	237.2400	12,979.6816	-0.2222	1.0088	513	236.1383	12,889.6699	234.3817	12,651.7744
458	236.6117	12,858.2646	237.5317	13,116.2813	-0.1398	1.0201	514	236.2650	12,897.0996	232.9767	13,903.8662
459	235.4850	12,876.5742	237.6267	12,972.5615	-0.3293	1.0075	515	235.9767	12,927.8047	233.0200	12,513.5488
460	235.7917	12,955.9707	238.1883	13,020.1982	-0.3643	1.0055	516	235.9050	12,896.3008	236.4300	13,311.4141
461	236.1467	12,865.3486	236.9600	12,792.7598	-0.1244	1.0057	517	235.8217	12,947.6455	232.0167	12,899.5527
462	235.7333	12,858.1201	238.7000	13,985.0449	-0.4435	1.0876	518	235.9667	12,851.8252	240.0833	13,812.7480
463	235.9733	12,935.6396	240.0950	13,305.9746	-0.6232	1.0286	519	235.7400	12,887.9551	235.4617	13,011.5029
464	235.7867	12,955.9707	240.1200	12,874.5830	-0.6604	1.0063	520	236.0967	12,905.9072	225.9450	11,915.5830
465	236.0950	12,873.2217	233.6433	12,556.0264	0.3766	1.0253	521	235.7550	12,887.9092	243.0500	13,566.0879
466	235.9767	12,936.9785	234.7150	12,761.8184	0.1928	1.0137	522	235.8600	13,032.5420	239.9350	13,773.7920
467	236.0117	12,955.1260	233.3817	12,616.7793	0.4029	1.0268	523	236.1100	12,825.7617	234.7167	13,348.6406
468	235.9200	12,990.8818	236.5783	12,799.5381	-0.1004	1.0149	524	236.3683	12,949.3906	237.0433	13,556.3584
469	236.1367	12,876.2920	232.9533	13,160.6387	0.4832	1.0221	525	235.8567	12,881.1973	239.0733	12,372.0215
470	235.8467	12,879.3223	233.4117	13,230.7119	0.3691	1.0273	526	235.9300	12,970.4180	231.9733	12,970.0957
471	236.0083	12,917.5439	236.2783	12,681.7793	-0.0413	1.0186	527	236.1483	12,883.8008	230.0750	13,295.5205
472	235.3967	12,976.9785	233.5017	13,402.8027	0.2858	1.0328	528	236.0650	12,904.5635	233.7883	12,756.2891
473	235.6200	12,913.0781	238.8833	13,256.7275	-0.4941	1.0266	529	235.9233	12,960.3838	236.3533	12,663.0205
474	235.9900	12,886.5127	240.1167	13,012.3535	-0.6281	1.0098	530	236.0467	12,846.3818	238.5217	12,632.0479
475	236.3050	12,883.1611	236.4083	13,422.0967	-0.0156	1.0418	531	235.6900	12,953.4092	238.3633	13,278.1387
476	235.9633	12,931.6455	237.3283	12,924.8369	-0.2079	1.0005	532	235.8283	12,837.6807	243.1167	12,796.7539
477	236.0167	12,930.1582	233.5083	13,685.3135	0.3766	1.0584	533	235.8633	12,863.5723	235.8817	12,863.2979
478	235.8767	12,842.4883	232.6850	14,187.1689	0.4746	1.0962	534	236.1450	12,901.1270	238.0767	12,744.5820
479	236.3567	12,889.7236	240.0367	13,363.8848	-0.5563	1.0368	535	235.7383	12,892.7754	238.2900	12,880.7910
480	236.4600	12,938.6621	239.5500	13,578.6367	-0.4648	1.0495	536	235.7267	12,918.0684	236.7300	12,972.1943
481	236.1917	12,960.0713	240.5017	13,800.7100	-0.6454	1.0649	537	236.3350	12,915.4004	235.8783	13,176.2188
482	236.0200	12,915.9570	235.8050	12,678.4590	0.0329	1.0187	538	235.6250	13,043.8496	234.6183	12,794.4414
483	236.2617	12,890.5010	237.0767	12,873.8506	-0.1244	1.0013	539	235.5500	12,888.6465	240.6850	13,121.1729
484	235.6067	12,932.9131	236.6817	13,307.1006	-0.1625	1.0273	540	235.9067	12,841.3770	231.0350	12,453.9785
485	235.8783	12,926.2529	237.3550	12,990.7754	-0.2247	1.0050	541	236.0617	12,931.2617	235.6567	12,892.8770
486	236.0950	12,841.1309	235.8683	12,511.3281	0.0349	1.0254	542	235.7117	12,895.4736	232.3350	13,711.3799
487	235.7833	12,948.5977	239.7050	12,699.2588	-0.5998	1.0196	543	235.8950	12,912.4209	232.3850	12,977.8486
488	236.0867	12,953.4355	235.1517	12,523.5127	0.1435	1.0343	544	235.9883	12,915.2549	235.2067	13,037.3594
489	235.6850	12,874.1465	237.2433	13,521.1826	-0.2349	1.0503	545	235.6883	12,926.4365	234.7200	12,931.0469
490	235.4850	12,834.5107	232.8667	12,950.0986	0.3994	1.0090	546	235.7800	12,966.9629	238.9317	12,792.2256
491	235.9950	12,897.4521	232.6600	12,649.0332	0.5111	1.0196	547	235.9317	12,911.9980	232.7350	13,142.3438
492	235.5867	12,930.7070	236.9800	12,588.8682	-0.2136	1.0272	548	235.8600	12,951.0869	232.8600	13,553.0752
493	236.4133	12,803.9854	238.2367	12,591.3232	-0.2803	1.0169	549	236.2267	12,852.2217	240.9217	12,348.1240
494	235.8517	12,952.6182	231.2233	12,101.5723	0.7162	1.0703	550	236.3150	12,793.0283	241.8383	14,040.7480
495	235.9267	12,872.6973	236.9417	12,594.1299	-0.1558	1.0221	551	236.2200	13,065.1729	238.4600	12,828.1885
496	236.1100	12,924.3965	242.1850	12,945.1348	-0.9252	1.0016	552	236.0233	12,990.3340	232.8550	12,568.5498
497	236.3100	12,889.1914	232.1733	13,267.6826	0.6265	1.0294	553	235.8467	12,905.4834	232.9733	12,668.2334
498	236.3617	12,930.8467	231.3817	13,631.5566	0.7485	1.0542	554	235.8117	12,941.7969	235.9900	13,026.5313
499	236.4233	12,838.4922	239.3100	13,264.4785	-0.4376	1.0332	555	235.8183	12,862.5078	234.8833	13,420.1934
500	236.1433	12,900.8281	239.8633	13,028.5420	-0.5659	1.0099	556	235.9283	12,877.2969	232.7467	12,674.6270
501	236.2233	12,915.9961	240.2133	12,970.7119	-0.6075	1.0042	557	235.6283	12,878.2881	233.0067	12,656.6943
502	236.0967	12,927.4941	237.3217	12,585.8896	-0.1879	1.0271	558	235.6983	13,042.9434	235.5117	13,032.4961
503	236.2433	12,853.6172	237.5833	13,049.1279	-0.2039	1.0152	559	235.9333	12,835.2275	241.8533	13,281.3203
504	235.8150	12,936.7031	234.8983	12,937.4951	0.1396	1.0001	560	235.8933	12,981.5059	241.8367	13,377.4824

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่ผู้สมัครเลือกในการสมัครเรียนต่อในระดับปริญญาตรีที่มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

การวิจัยด้วยวิธีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

MMS						SRS						MMS						SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value			
561	235.8583	12.886.1680	241.7733	13,310.0918	-0.8952	1.0329	617	235.8517	12.897.0869	234.9067	13,062.1084	0.1437	1.0128	617	235.8517	12.897.0869	234.9067	13,062.1084	0.1437	1.0128			
562	235.6233	12.975.7598	240.8033	12,942.8994	-0.7881	1.0025	618	235.6350	12.841.1211	239.1650	13,008.6709	-0.5378	1.0130	618	235.6350	12.841.1211	239.1650	13,008.6709	-0.5378	1.0130			
563	236.2300	12.865.9717	232.9450	12,114.5645	0.5091	1.0620	619	236.0750	12.860.8496	239.1150	12,652.9248	-0.4662	1.0164	619	236.0750	12.860.8496	239.1150	12,652.9248	-0.4662	1.0164			
564	235.8483	12.931.4746	237.3850	12,626.2119	-0.2345	1.0082	620	236.0700	12.963.7539	232.4367	13,106.2461	0.5512	1.0110	620	236.0700	12.963.7539	232.4367	13,106.2461	0.5512	1.0110			
565	235.9383	12.950.0771	234.4350	12,211.0557	0.2321	1.0605	621	235.9800	12.873.4854	236.0117	12,827.2051	-0.0048	1.0036	621	235.9800	12.873.4854	236.0117	12,827.2051	-0.0048	1.0036			
566	236.1250	12.953.3564	234.4050	12,722.5537	0.2629	1.0181	622	236.1850	12.875.1934	237.1650	13,233.2178	-0.1561	1.0278	622	236.1850	12.875.1934	237.1650	13,233.2178	-0.1561	1.0278			
567	235.6067	12.945.4063	234.6150	12,989.6582	0.1508	1.0034	623	235.8683	12.960.5176	235.3217	12,706.2305	0.0836	1.0200	623	235.8683	12.960.5176	235.3217	12,706.2305	0.0836	1.0200			
568	235.5067	12.975.0176	242.0800	13,937.7568	-0.9815	1.0742	624	236.2867	12.899.7852	238.1017	13,487.2549	-0.2737	1.0455	624	236.2867	12.899.7852	238.1017	13,487.2549	-0.2737	1.0455			
569	236.3250	12.921.9512	237.4167	12,292.4336	-0.1684	1.0512	625	236.2917	12.999.0527	237.0567	12,232.4238	-0.1181	1.0578	625	236.2917	12.999.0527	237.0567	12,232.4238	-0.1181	1.0578			
570	235.9750	12.817.3252	238.3233	13,063.3076	-0.3576	1.0192	626	235.7217	12.895.8057	233.4283	13,898.6113	0.3464	1.0390	626	235.7217	12.895.8057	233.4283	13,898.6113	0.3464	1.0390			
571	235.9500	12.938.7031	237.1900	12,780.7588	-0.1894	1.0124	627	236.4483	12.854.0820	237.5417	13,214.0732	-0.1659	1.0280	627	236.4483	12.854.0820	237.5417	13,214.0732	-0.1659	1.0280			
572	235.4183	12.946.8184	230.6100	12,477.6777	0.7387	1.0376	628	235.8867	13.006.3848	235.0783	12,730.3740	0.1234	1.0217	628	235.8867	13.006.3848	235.0783	12,730.3740	0.1234	1.0217			
573	236.0050	12.897.2939	232.6050	12,832.5146	0.5192	1.0050	629	235.6633	13.018.0557	237.4283	11,888.3740	-0.2739	1.0950	629	235.6633	13.018.0557	237.4283	11,888.3740	-0.2739	1.0950			
574	235.8467	12.944.5273	241.1800	13,244.4150	-0.8073	1.0232	630	236.0700	12.865.3555	230.6983	13,314.7832	0.8132	1.0349	630	236.0700	12.865.3555	230.6983	13,314.7832	0.8132	1.0349			
575	235.9767	12.903.3633	234.9367	13,065.8193	0.1581	1.0126	631	235.7817	12.881.5146	235.9950	13,197.8760	-0.0324	1.0246	631	235.7817	12.881.5146	235.9950	13,197.8760	-0.0324	1.0246			
576	235.7833	13.006.5234	237.8600	13,181.3359	-0.3143	1.0134	632	236.1100	12.834.7256	237.9167	13,231.8262	-0.2741	1.0309	632	236.1100	12.834.7256	237.9167	13,231.8262	-0.2741	1.0309			
577	236.1900	12.898.1826	234.3733	12,688.3076	0.2782	1.0165	633	236.1100	12.916.6816	236.3733	12,958.1172	-0.0401	1.0032	633	236.1100	12.916.6816	236.3733	12,958.1172	-0.0401	1.0032			
578	235.9000	12.909.6133	242.1917	12,512.2900	-0.9666	1.0318	634	236.1033	12.861.3027	239.8383	12,810.6650	-0.5710	1.0040	634	236.1033	12.861.3027	239.8383	12,810.6650	-0.5710	1.0040			
579	235.6400	12.890.6748	229.4767	13,140.0127	0.9357	1.0193	635	235.7350	12.864.0996	233.3050	13,256.2578	0.3683	1.0305	635	235.7350	12.864.0996	233.3050	13,256.2578	0.3683	1.0305			
580	236.0450	12.764.8262	236.2317	13,511.1582	-0.0282	1.0585	636	236.1633	12.951.4971	239.7517	13,128.1787	-0.5443	1.0136	636	236.1633	12.951.4971	239.7517	13,128.1787	-0.5443	1.0136			
581	236.2267	12.892.1904	234.8450	12,899.8594	0.2107	1.0006	637	235.8017	12.841.0537	231.0433	13,530.8359	0.7177	1.0537	637	235.8017	12.841.0537	231.0433	13,530.8359	0.7177	1.0537			
582	236.0617	12.931.2188	237.8267	13,207.5156	-0.2674	1.0214	638	236.1400	13.001.7070	239.1783	13,222.6826	-0.4596	1.0170	638	236.1400	13.001.7070	239.1783	13,222.6826	-0.4596	1.0170			
583	236.4100	12.896.7354	237.0533	12,551.4365	-0.0988	1.0275	639	236.3333	12.887.8662	230.8783	12,498.8262	0.8386	1.0311	639	236.3333	12.887.8662	230.8783	12,498.8262	0.8386	1.0311			
584	235.6833	12.935.0850	234.0517	13,255.1025	0.2470	1.0247	640	236.1650	12.967.8340	239.2383	12,917.1221	-0.4679	1.0039	640	236.1650	12.967.8340	239.2383	12,917.1221	-0.4679	1.0039			
585	235.8000	12.911.8633	236.5167	13,122.9414	-0.1088	1.0171	641	235.6283	12.994.3252	238.4300	13,091.4678	-0.4249	1.0075	641	235.6283	12.994.3252	238.4300	13,091.4678	-0.4249	1.0075			
586	236.0383	12.977.7803	237.2483	12,564.0215	-0.1855	1.0329	642	236.1383	12.955.7549	232.8783	13,181.7715	0.4939	1.0174	642	236.1383	12.955.7549	232.8783	13,181.7715	0.4939	1.0174			
587	235.7800	12.969.6670	237.2317	13,429.7586	-0.2189	1.0355	643	236.4733	12.904.6582	237.6950	12,534.5645	-0.1876	1.0235	643	236.4733	12.904.6582	237.6950	12,534.5645	-0.1876	1.0235			
588	236.0817	12.975.8975	239.6167	13,663.8662	-0.5305	1.0529	644	236.0933	12.978.8965	238.5817	13,126.8135	-0.3772	1.0114	644	236.0933	12.978.8965	238.5817	13,126.8135	-0.3772	1.0114			
589	235.4367	12.868.8457	238.6967	12,945.9775	-0.4970	1.0060	645	236.3300	12.936.1523	234.5617	12,576.1914	0.2712	1.0286	645	236.3300	12.936.1523	234.5617	12,576.1914	0.2712	1.0286			
590	235.7500	12.944.0781	241.4950	12,727.1504	-0.8783	1.0170	646	236.3100	13.019.8867	231.7483	12,771.5430	0.6958	1.0194	646	236.3100	13.019.8867	231.7483	12,771.5430	0.6958	1.0194			
591	235.4183	12.935.5020	239.7933	12,866.0273	-0.6679	1.0054	647	236.2334	12.954.8652	236.6633	13,545.2021	-0.0647	1.0456	647	236.2334	12.954.8652	236.6633	13,545.2021	-0.0647	1.0456			
592	235.6967	12.929.5635	234.8517	12,520.8760	0.1297	1.0326	648	235.5883	12.939.8252	237.5417	12,976.4307	-0.2972	1.0029	648	235.5883	12.939.8252	237.5417	12,976.4307	-0.2972	1.0029			
593	235.6283	12.837.3379	235.4183	12,818.5322	0.0627	1.0015	649	235.9750	12.953.2266	238.9517	13,009.3428	-0.4525	1.0043	649	235.9750	12.953.2266	238.9517	13,009.3428	-0.4525	1.0043			
594	235.9633	12.905.7959	237.0067	13,194.8711	-0.1582	1.0224	650	236.5067	12.932.6045	237.7167	12,589.4551	-0.1855	1.0273	650	236.5067	12.932.6045	237.7167	12,589.4551	-0.1855	1.0273			
595	236.2150	12.891.4258	233.1283	12,449.1289	0.4750	1.0355	651	235.8567	12.943.6650	238.0867	12,826.8838	-0.3403	1.0091	651	235.8567	12.943.6650	238.0867	12,826.8838	-0.3403	1.0091			
596	236.1767	12.937.2744	238.4200	12,888.3311	-0.3419	1.0038	652	236.0883	12.905.5576	237.9717	13,117.1475	-0.2893	1.0311	652	236.0883	12.905.5576	237.9717	13,117.1475	-0.2893	1.0311			
597	235.6167	12.854.4043	235.8717	13,315.8564	-0.0383	1.0359	653	235.7983	12.995.5303	239.2367	12,833.1260	-0.5241	1.0127	653	235.7983	12.995.5303	239.2367	12,833.1260	-0.5241	1.0127			
598	235.7500	12.977.1875	230.7950	12,905.5576	0.7544	1.0056	654	235.8433	12.861.8359	235.2617	13,152.1484	0.0883	1.0226	654	235.8433	12.861.8359	235.2617	13,152.1484	0.0883	1.0226			
599	236.0750	12.918.1914	238.8433	13,053.3047	-0.4208	1.0105	655	235.9350	12.823.5156	236.0533	12,355.2559	-0.0182	1.0460	655	235.9350	12.823.5156	236.0533	12,355.2559	-0.0182	1.0460			
600	236.0367	12.795.7061	236.4450	12,322.2188	-0.0631	1.0384	656	236.3417	12.839.3682	232.8717	13,056.2070	0.5282	1.0169</										

ตาราง 59 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่ผู้เรียนได้จากการเรียนแบบวิธีการเรียนด้วยคอมพิวเตอร์กับเทคนิคการเรียนแบบปกติ

MMS							SRS							MMS							SRS						
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	t-value	F-value		
673	236.0033	12.857.7354	241.8817	12.522.5840	-0.9038	1.0268	729	235.8783	12.888.7529	241.5300	12.949.2207	-0.8612	1.0047	729	235.8783	12.888.7529	241.5300	12.949.2207	-0.8612	1.0047	729	235.8783	12.888.7529	-0.8612	1.0047		
674	236.7717	12.929.2891	237.3400	12.639.6641	-0.2402	1.0229	730	235.6567	12.928.6416	231.8450	12.384.4902	0.5866	1.0439	730	235.6567	12.928.6416	231.8450	12.384.4902	0.5866	1.0439	730	235.6567	12.928.6416	0.5866	1.0439		
675	236.0300	12.986.8682	239.1950	12.780.9600	-0.4830	1.0161	731	235.9600	12.913.5889	237.9233	12.803.1992	-0.2999	1.0086	731	235.9600	12.913.5889	237.9233	12.803.1992	-0.2999	1.0086	731	235.9600	12.913.5889	-0.2999	1.0086		
676	236.1167	12.951.5645	238.6467	12.943.9883	-0.8851	1.0006	732	236.0100	12.890.9678	242.0833	12.780.3340	-0.9285	1.0087	732	236.0100	12.890.9678	242.0833	12.780.3340	-0.9285	1.0087	732	236.0100	12.890.9678	-0.9285	1.0087		
677	236.0800	12.903.3301	238.0633	12.430.6807	-0.3052	1.0380	733	235.6300	12.899.4268	236.9033	12.814.3809	-0.1941	1.0144	733	235.6300	12.899.4268	236.9033	12.814.3809	-0.1941	1.0144	733	235.6300	12.899.4268	-0.1941	1.0144		
678	236.6033	12.924.2871	233.5567	13.354.8086	0.3093	1.0333	734	235.9800	12.962.0371	234.1850	12.763.3916	0.2741	1.0156	734	235.9800	12.962.0371	234.1850	12.763.3916	0.2741	1.0156	734	235.9800	12.962.0371	0.2741	1.0156		
679	236.0733	12.912.8389	235.2667	12.903.2813	0.1230	1.0007	735	235.8150	12.896.2461	236.2783	13.039.6152	-0.0705	1.0111	735	235.8150	12.896.2461	236.2783	13.039.6152	-0.0705	1.0111	735	235.8150	12.896.2461	-0.0705	1.0111		
680	236.8033	12.846.2246	236.1833	12.884.1797	-0.0573	1.0030	736	236.4750	12.908.8291	235.0133	13.103.5596	0.2220	1.0151	736	236.4750	12.908.8291	235.0133	13.103.5596	0.2220	1.0151	736	236.4750	12.908.8291	0.2220	1.0151		
681	236.1783	12.928.7246	238.6500	13.273.4336	-0.3740	1.0267	737	236.1717	12.872.3838	235.4917	12.774.8330	0.1040	1.0076	737	236.1717	12.872.3838	235.4917	12.774.8330	0.1040	1.0076	737	236.1717	12.872.3838	0.1040	1.0076		
682	236.4683	12.981.5859	241.2550	12.440.3252	-0.8890	1.0435	738	236.2350	12.907.7617	232.3283	12.924.3926	0.5954	1.0013	738	236.2350	12.907.7617	232.3283	12.924.3926	0.5954	1.0013	738	236.2350	12.907.7617	0.5954	1.0013		
683	236.3133	12.880.0908	231.2967	13.275.9053	0.7598	1.0307	739	235.9717	12.927.0205	239.4117	12.858.7852	-0.5247	1.0053	739	235.9717	12.927.0205	239.4117	12.858.7852	-0.5247	1.0053	739	235.9717	12.927.0205	-0.5247	1.0053		
684	236.2750	12.944.1670	231.7500	12.987.0625	0.6883	1.0033	740	235.9583	12.884.0527	231.8833	12.305.1953	0.6289	1.0470	740	235.9583	12.884.0527	231.8833	12.305.1953	0.6289	1.0470	740	235.9583	12.884.0527	0.6289	1.0470		
685	236.8550	12.891.4072	237.9117	13.316.9268	-0.3112	1.0330	741	235.5917	13.029.7275	237.6300	12.547.0752	-0.3122	1.0385	741	235.5917	13.029.7275	237.6300	12.547.0752	-0.3122	1.0385	741	235.5917	13.029.7275	-0.3122	1.0385		
686	236.8650	12.891.4160	232.6533	13.230.0430	0.4862	1.0309	742	236.1483	12.933.7900	233.8083	13.072.6709	0.3554	1.0107	742	236.1483	12.933.7900	233.8083	13.072.6709	0.3554	1.0107	742	236.1483	12.933.7900	0.3554	1.0107		
687	236.7700	13.010.2617	234.5667	12.564.6064	0.1843	1.0355	743	236.1300	12.896.5420	239.3267	13.058.2842	-0.4880	1.0125	743	236.1300	12.896.5420	239.3267	13.058.2842	-0.4880	1.0125	743	236.1300	12.896.5420	-0.4880	1.0125		
688	236.1167	12.903.4678	234.6350	12.853.6133	0.2261	1.0039	744	236.0817	12.893.7441	236.8850	13.623.8301	-0.4217	1.0566	744	236.0817	12.893.7441	236.8850	13.623.8301	-0.4217	1.0566	744	236.0817	12.893.7441	-0.4217	1.0566		
689	236.4833	12.854.4678	234.6350	12.863.4492	0.1296	1.0007	745	236.1667	12.871.1846	237.9517	12.758.0215	-0.2731	1.0089	745	236.1667	12.871.1846	237.9517	12.758.0215	-0.2731	1.0089	745	236.1667	12.871.1846	-0.2731	1.0089		
690	236.1167	12.963.4824	232.7883	12.510.1826	0.5108	1.0362	746	235.9833	12.882.3291	232.3967	12.975.5820	0.5463	1.0072	746	235.9833	12.882.3291	232.3967	12.975.5820	0.5463	1.0072	746	235.9833	12.882.3291	0.5463	1.0072		
691	236.8283	12.832.3008	236.4467	13.760.1738	-0.0929	1.0728	747	236.0133	12.798.6094	241.0367	12.626.6836	-0.7717	1.0136	747	236.0133	12.798.6094	241.0367	12.626.6836	-0.7717	1.0136	747	236.0133	12.798.6094	-0.7717	1.0136		
692	236.8633	12.975.1309	235.9217	13.339.0586	-0.0088	1.0280	748	236.2900	12.893.2598	236.2333	12.461.1143	-0.2990	1.0347	748	236.2900	12.893.2598	236.2333	12.461.1143	-0.2990	1.0347	748	236.2900	12.893.2598	-0.2990	1.0347		
693	236.5833	12.956.6934	236.0483	13.532.1650	-0.0730	1.0444	749	236.1117	12.937.3555	235.2667	12.864.5430	0.1289	1.0057	749	236.1117	12.937.3555	235.2667	12.864.5430	0.1289	1.0057	749	236.1117	12.937.3555	0.1289	1.0057		
694	236.5750	12.964.5381	236.3767	12.966.0010	-0.1219	1.0001	750	236.4100	12.918.0654	236.7683	12.593.1650	-0.0549	1.0258	750	236.4100	12.918.0654	236.7683	12.593.1650	-0.0549	1.0258	750	236.4100	12.918.0654	-0.0549	1.0258		
695	236.2383	12.969.0630	237.6333	12.564.0654	-0.2444	1.0338	751	236.1917	12.945.7949	234.3333	12.549.6387	0.2851	1.0316	751	236.1917	12.945.7949	234.3333	12.549.6387	0.2851	1.0316	751	236.1917	12.945.7949	0.2851	1.0316		
696	236.9700	12.931.1295	236.6333	12.950.2959	-0.1010	1.0015	752	236.0950	12.834.2607	242.4850	12.678.2617	-0.9799	1.0123	752	236.0950	12.834.2607	242.4850	12.678.2617	-0.9799	1.0123	752	236.0950	12.834.2607	-0.9799	1.0123		
697	236.0150	13.023.3086	237.3517	12.763.2451	-0.2039	1.0204	753	236.8650	12.884.6621	239.4967	12.929.7598	-0.5521	1.0035	753	236.8650	12.884.6621	239.4967	12.929.7598	-0.5521	1.0035	753	236.8650	12.884.6621	-0.5521	1.0035		
698	236.8517	12.955.3027	235.5500	13.772.4453	0.0497	1.0631	754	236.3100	12.938.2344	235.2100	12.874.7402	0.1677	1.0049	754	236.3100	12.938.2344	235.2100	12.874.7402	0.1677	1.0049	754	236.3100	12.938.2344	0.1677	1.0049		
699	236.8183	12.835.9844	243.2683	12.478.6758	-1.1470	1.0286	755	236.5533	12.927.4531	231.0250	12.924.8105	0.8422	1.0002	755	236.5533	12.927.4531	231.0250	12.924.8105	0.8422	1.0002	755	236.5533	12.927.4531	0.8422	1.0002		
700	236.7150	12.877.6055	234.5133	12.126.5879	0.1861	1.0619	756	236.5450	12.916.7324	234.7200	13.384.2021	0.1246	1.0362	756	236.5450	12.916.7324	234.7200	13.384.2021	0.1246	1.0362	756	236.5450	12.916.7324	0.1246	1.0362		
701	236.6400	12.872.6133	233.4450	12.253.4609	0.3392	1.0505	757	236.7900	12.904.4453	235.2867	12.504.9326	0.0773	1.0319	757	236.7900	12.904.4453	235.2867	12.504.9326	0.0773	1.0319	757	236.7900	12.904.4453	0.0773	1.0319		
702	236.3483	12.861.5590	226.5783	12.286.1162	1.5091	1.0469	758	236.8917	12.985.6357	234.0017	12.819.3721	0.2882	1.0130	758	236.8917	12.985.6357	234.0017	12.819.3721	0.2882	1.0130	758	236.8917	12.985.6357	0.2882	1.0130		
703	236.1383	12.910.2725	232.6050	13.353.2227	0.5341	1.0343	759	236.1767	12.895.4209	237.9150	12.301.4814	-0.2682	1.0483	759	236.1767	12.895.4209	237.9150	12.301.4814	-0.2682	1.0483	759	236.1767	12.895.4209	-0.2682	1.0483		
704	236.1017	12.894.4238	230.5267	12.559.4686	0.8559	1.0267	760	236.1933	12.867.8721	234.7117	12.950.9316	0.2259	1.0065	760	236.1933	12.867.8721	234.7117	12.950.9316	0.2259	1.0065	760	236.1933	12.867.8721	0.2259	1.0065		
705	236.9533	12.835.5689	240.0933	12.759.9111	-0.6339																						

ตาราง 50 (ต่อ) เปรียบเทียบค่าเฉลี่ยและค่าความแปรปรวนของคะแนนที่ผู้โดยเทคนิคการสุ่มแบบอยุ่ใจถึงพหุคูณกับเทคนิคการสุ่มแบบง่าย

กรณีข้อมูลมีการแจกแจงแบบปกติ จำนวนกลุ่มตัวอย่าง เท่ากับ 600 คน

SRS							MMS						
MMS			SRS				MMS			SRS			
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value
785	235.7250	12,949.8711	233.5117	12,875.6543	0.3374	1.0058	841	235.8883	12,945.3184	233.8783	13,092.5459	0.3051	1.0114
786	236.1133	12,894.0078	239.1033	13,415.9160	-0.4515	1.0405	842	235.5950	13,031.7256	232.0283	13,394.2559	0.5374	1.0278
787	236.2633	12,906.8408	236.6600	12,502.1279	-0.0610	1.0324	843	236.2150	12,903.9688	233.1033	12,546.5273	0.4778	1.0285
788	236.3683	12,876.9389	243.0250	13,622.1865	-1.0017	1.0579	844	236.2200	13,027.1484	230.1783	12,983.6943	0.9176	1.0033
789	236.3317	12,901.8926	233.3200	13,042.7285	0.4580	1.0109	845	235.9600	12,898.6787	234.7067	12,807.2725	0.1915	1.0071
790	235.9984	12,961.9414	230.3817	12,891.9609	0.8556	1.0054	846	235.9033	12,851.2070	237.2033	12,781.4277	-0.1989	1.0055
791	235.8633	12,925.5537	237.1350	13,045.2471	-0.1949	1.0093	847	235.9000	12,941.7256	242.8350	13,236.4170	-1.0499	1.0228
792	235.2200	12,877.5557	241.8683	12,042.4131	-1.0331	1.0635	848	235.9167	12,871.3271	234.6400	12,928.4551	0.1642	1.0044
793	235.8400	12,970.3184	233.6533	12,840.7813	0.3334	1.0101	849	236.1200	12,891.1768	236.7833	12,543.9600	-0.1019	1.0277
794	236.0267	12,948.9785	234.7917	12,836.5342	0.1884	1.0088	850	236.0067	12,887.1240	238.4217	12,799.2607	-0.3691	1.0069
795	236.0650	12,943.5283	230.7650	12,399.8076	0.8155	1.0438	851	236.0633	12,881.1260	233.0617	12,918.9248	0.4578	1.0029
796	236.0283	12,817.5420	240.5183	12,721.8477	-0.6882	1.0075	852	236.0283	12,762.4297	247.4250	12,799.5020	-1.7614	1.0029
797	235.7083	12,950.6914	239.6383	12,472.2100	-0.6037	1.0384	853	235.7767	12,939.9102	232.7333	12,882.1523	0.4639	1.0045
798	236.2333	12,972.7764	238.3983	13,075.9346	-0.3286	1.0080	854	236.3067	12,924.3271	231.7667	12,770.5088	0.6938	1.0120
799	236.1700	12,879.0234	235.5217	13,410.8760	0.0979	1.0413	855	236.0933	12,931.0283	236.5800	12,781.0049	-0.0743	1.0117
800	236.2783	12,868.9707	235.1967	13,200.1885	0.1643	1.0305	856	236.0100	12,902.1328	235.8317	12,949.1348	0.1033	1.0036
801	236.0867	12,959.1670	238.1017	13,032.2764	-0.3062	1.0056	857	236.2200	12,904.7002	231.8450	13,054.8877	0.6651	1.0116
802	236.0050	12,863.2949	235.1250	13,381.6436	0.1331	1.0403	858	235.5967	12,879.1689	239.8833	12,242.4238	-0.5852	1.0520
803	236.5200	12,895.1357	230.4300	13,510.0654	0.9180	1.0477	859	235.9050	12,948.4033	232.6117	12,877.2080	0.5020	1.0055
804	235.7500	12,962.7930	238.8450	13,294.6572	-0.4679	1.0256	860	235.9733	12,908.8291	237.0633	13,399.0547	-0.1646	1.0380
805	236.2383	12,772.6191	231.8033	13,231.8340	0.6737	1.0360	861	236.0233	13,015.1582	238.1617	12,918.8057	-0.3253	1.0075
806	236.1300	12,911.8963	236.7500	12,941.8936	-0.0945	1.0023	862	235.8067	12,968.8398	237.9033	12,490.1279	-0.3219	1.0383
807	236.0067	12,938.9398	233.2517	12,352.5742	0.4243	1.0474	863	236.5183	13,027.6719	239.7950	12,623.3301	-0.5011	1.0320
808	235.7950	12,989.0039	232.7950	13,128.4424	0.4549	1.0124	864	236.1167	12,875.9580	236.3567	13,176.6807	-0.3399	1.0234
809	236.0967	12,950.5645	233.5467	12,390.7324	0.3924	1.0452	865	236.1133	12,898.3281	235.5367	12,397.7813	0.0888	1.0404
810	235.7683	12,892.9092	242.0700	14,108.1689	-0.9394	1.0943	866	236.0150	13,005.1035	235.6583	12,528.6113	0.0547	1.0380
811	236.2367	12,896.1885	235.6267	12,778.2246	0.0933	1.0082	867	236.0650	12,875.4375	233.9300	12,775.5674	0.3265	1.0078
812	235.9467	12,989.8535	240.7417	12,634.5879	-0.7337	1.0281	868	236.1183	13,031.1777	239.2517	13,363.6465	-0.4724	1.0255
813	236.2850	12,985.0947	234.1850	12,585.7422	0.3217	1.0317	869	235.8300	12,917.1943	233.7300	12,852.8223	0.3198	1.0028
814	236.2383	12,915.9844	234.3267	12,775.4658	0.2921	1.0110	870	235.8533	12,863.9053	234.6017	12,456.4858	0.1927	1.0327
815	236.0733	12,957.8301	237.0783	13,286.8818	-0.1520	1.0254	871	235.4917	12,879.4717	242.2017	13,355.1709	-1.0148	1.0369
816	235.5650	12,954.5908	237.8050	13,465.4648	-0.3376	1.0394	872	235.8517	12,970.8848	238.5917	12,629.3320	-0.4195	1.0270
817	235.4433	12,781.7813	234.2767	13,190.0264	0.1775	1.0272	873	235.7250	12,978.9082	238.2867	12,773.9072	-0.3910	1.0160
818	235.6417	12,962.9600	236.9267	13,470.1445	-0.1936	1.0391	874	235.7633	12,998.1758	231.4233	13,012.1611	0.6592	1.0011
819	236.1817	12,898.7900	231.4117	12,362.5313	0.7351	1.0434	875	235.6533	12,831.7100	234.4983	12,916.0186	0.1763	1.0066
820	236.4083	12,988.3340	235.3783	13,187.3662	0.1562	1.0232	876	236.0700	12,911.8789	231.2967	12,871.2578	0.7282	1.0032
821	236.6200	12,955.2637	235.8817	13,428.0127	-0.0395	1.0365	877	236.6300	12,949.6736	232.0217	12,447.1777	0.7083	1.0404
822	236.0683	12,855.2676	230.7583	13,122.4961	0.8070	1.0208	878	235.8133	12,933.5352	232.0400	13,304.0117	0.5706	1.0286
823	236.0733	12,935.4805	234.4417	13,310.2354	0.2467	1.0290	879	236.3233	12,947.2852	236.7000	13,143.3223	-0.0571	1.0151
824	235.8050	12,906.7617	235.5333	12,428.9570	0.0418	1.0384	880	235.7267	12,957.2148	237.2833	13,031.5859	-0.2365	1.0057
825	236.3633	12,891.1553	236.2550	13,156.7295	0.0164	1.0206	881	235.8883	12,866.8291	239.5867	13,301.1445	-0.5600	1.0338
826	235.6417	13,026.5283	236.7133	13,358.3955	-0.1616	1.0255	882	236.4167	12,946.0020	235.7717	12,994.2314	0.0981	1.0037
827	236.0717	12,929.0303	234.8900	12,179.8135	0.1827	1.0616	883	236.3900	12,986.4922	239.0517	13,310.1240	-0.4021	1.0249
828	236.2267	12,952.2168	238.9617	12,931.2402	-0.4164	1.0016	884	235.8550	12,827.0098	231.0550	12,915.2061	0.7328	1.0069
829	235.8567	12,939.5879	233.8050	12,617.5713	0.3144	1.0255	885	235.9250	12,915.0898	231.4117	12,978.8037	0.6871	1.0045
830	235.8967	12,909.9648	230.1317	12,904.1094	0.8789	1.0005	886	236.0350	12,847.6631	234.0367	12,677.7080	0.3064	1.0134
831	235.9767	13,031.2256	238.6817	14,119.4473	-0.4021	1.0335	887	236.0650	12,885.0029	230.8350	13,185.0645	0.7934	1.0233
832	236.3150	12,910.2354	233.8683	13,319.9424	0.3700	1.0317	888	236.1250	12,939.3477	236.1150	12,589.7832	0.0015	1.0278
833	235.7817	12,907.6123	239.3317	13,176.2402	-0.5384	1.0208	889	236.0750	12,909.8828	236.9000	13,676.1035	-0.1239	1.0594
834	236.3650	12,969.1514	234.3833	13,231.8857	0.2999	1.0203	890	235.9800	12,899.2852	239.2150	12,342.7412	-0.4988	1.0451
835	236.0783	12,966.3633	232.0817	13,003.4453	0.6075	1.0029	891	236.2483	12,874.1602	235.0367	12,946.8369	0.1847	1.0056
836	235.9117	12,982.9932	232.4417	13,515.9785	0.5221	1.0411	892	235.8967	12,981.4824	240.0667	13,059.0674	-0.6330	1.0060
837	236.1567	13,000.6221	240.8583	12,712.2500	-0.7182	1.0227	893	235.9350	12,961.2070	238.9867	12,864.1064	-0.4651	1.0075
838	236.4033	12,999.6456	238.7967	12,884.1660	-0.3684	1.0449	894	236.0860	12,818.4561	235.4583	13,021.2285	0.0955	1.0158
839	236.2583	13,008.8828	231.1633	13,520.6074	0.7662	1.0398	895	236.0750	12,931.0439	239.5350	13,159.9365	-0.5247	1.0177
840	236.2133	12,799.5576	233.4150	13,416.9795	0.4233	1.0482	896	235.8400	12,870.5957	240.3883	12,481.0908	-0.6997	1.0312

แบบ 50 (ฉบับแก้ไขเพิ่มเติม) สำหรับตรวจสอบความแตกต่างของค่าเฉลี่ยระหว่างกลุ่มตัวอย่างที่มีการจับคู่กัน (Paired Sample t-test) โดยใช้โปรแกรมสำเร็จรูป
 ข้อมูลตัวอย่าง: จำนวนตัวอย่าง = 100 คู่

MMS						SRS						MMS						SRS					
No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value	No.	Mean	Variance	Mean	Variance	t-value	F-value			
897	235.9133	12.931.8916	234.6133	12.567.6836	0.1956	1.0492	953	236.2533	12.976.9902	241.0800	12.564.2402	-0.7398	1.0329	953	236.2533	12.976.9902	241.0800	12.564.2402	-0.7398	1.0329			
898	235.4350	12.857.5098	235.0100	12.952.5078	0.0648	1.0074	954	235.7017	12.844.3818	236.0867	12.796.2666	-0.0589	1.0038	954	235.7017	12.844.3818	236.0867	12.796.2666	-0.0589	1.0038			
899	235.6967	12.876.3682	236.1167	12.515.1387	-0.0646	1.0289	955	235.9533	12.967.8262	232.2817	12.588.3613	0.5626	1.0301	955	235.9533	12.967.8262	232.2817	12.588.3613	0.5626	1.0301			
900	236.2683	12.800.6943	238.5067	12.315.2148	-0.3460	1.0394	956	236.0600	12.908.4219	239.5933	12.695.7012	-0.5409	1.0168	956	236.0600	12.908.4219	239.5933	12.695.7012	-0.5409	1.0168			
901	235.5300	12.913.5479	236.9500	12.459.9775	-0.2184	1.0364	957	235.8767	12.915.6436	238.1217	12.587.8184	-0.3443	1.0260	957	235.8767	12.915.6436	238.1217	12.587.8184	-0.3443	1.0260			
902	235.7400	12.753.8604	234.8517	12.925.6379	0.1358	1.0135	958	235.8817	12.883.7832	231.2217	13.115.3594	0.7079	1.0180	958	235.8817	12.883.7832	231.2217	13.115.3594	0.7079	1.0180			
903	235.7033	12.859.7705	239.5833	12.943.9932	-0.5917	1.0065	959	235.7150	12.926.3555	234.8517	12.802.0117	0.1318	1.0097	959	235.7150	12.926.3555	234.8517	12.802.0117	0.1318	1.0097			
904	236.1933	12.911.3379	241.5967	12.801.4463	-0.8254	1.0086	960	235.0183	12.813.4785	237.3267	13.175.0518	-0.1988	1.0282	960	235.0183	12.813.4785	237.3267	13.175.0518	-0.1988	1.0282			
905	236.1763	12.944.6377	230.4850	13.488.8730	0.8578	1.0420	961	235.1567	13.002.8047	237.0433	13.396.8428	-0.1337	1.0305	961	235.1567	13.002.8047	237.0433	13.396.8428	-0.1337	1.0305			
906	236.0150	12.060.9180	232.9733	12.345.9824	0.4674	1.0579	962	235.8833	12.893.2412	231.7183	13.572.3877	0.6271	1.0527	962	235.8833	12.893.2412	231.7183	13.572.3877	0.6271	1.0527			
907	235.6400	12.933.6406	233.2967	12.405.6348	0.3606	1.0426	963	235.5700	12.930.0010	234.1617	13.157.4033	0.2136	1.0176	963	235.5700	12.930.0010	234.1617	13.157.4033	0.2136	1.0176			
908	236.5917	13.009.0508	240.3900	12.509.6836	-0.5824	1.0399	964	236.3267	12.973.9570	236.5533	13.125.9102	-0.0344	1.0117	964	236.3267	12.973.9570	236.5533	13.125.9102	-0.0344	1.0117			
909	235.9483	12.864.3496	239.6283	13.064.4492	-0.5598	1.0156	965	235.6067	13.045.0674	233.8850	13.153.6230	0.2605	1.0083	965	235.6067	13.045.0674	233.8850	13.153.6230	0.2605	1.0083			
910	235.7000	12.999.7979	237.0683	13.559.4180	-0.2057	1.0430	966	235.6700	12.938.1660	233.4317	12.934.2939	0.3409	1.0003	966	235.6700	12.938.1660	233.4317	12.934.2939	0.3409	1.0003			
911	236.1683	12.896.7715	231.2750	13.060.6484	0.7440	1.0127	967	235.2767	12.994.1357	235.8333	13.168.4063	-0.0843	1.0134	967	235.2767	12.994.1357	235.8333	13.168.4063	-0.0843	1.0134			
912	236.2483	12.887.2480	239.6050	13.227.0557	-0.5088	1.0264	968	235.7867	12.948.2539	228.6500	12.636.8789	1.0929	1.0246	968	235.7867	12.948.2539	228.6500	12.636.8789	1.0929	1.0246			
913	235.3050	12.876.9014	236.0117	13.434.1035	-0.1067	1.0433	969	236.2100	12.884.1602	238.2117	12.656.7969	-0.3068	1.0180	969	236.2100	12.884.1602	238.2117	12.656.7969	-0.3068	1.0180			
914	235.0000	12.932.3105	238.5167	13.289.6963	-0.3906	1.0237	970	236.0467	12.931.4307	235.3150	12.777.2599	0.1118	1.0121	970	236.0467	12.931.4307	235.3150	12.777.2599	0.1118	1.0121			
915	235.0650	13.040.3340	238.3800	12.795.0010	-0.3528	1.0192	971	236.0550	12.851.7861	232.8200	12.604.4648	0.4950	1.0204	971	236.0550	12.851.7861	232.8200	12.604.4648	0.4950	1.0204			
916	235.6000	12.950.1942	232.3000	13.006.5947	0.5017	1.0344	972	236.2067	12.889.7832	233.7200	13.962.4727	0.3717	1.0832	972	236.2067	12.889.7832	233.7200	13.962.4727	0.3717	1.0832			
917	235.7267	12.952.2148	238.2483	13.003.2989	-0.3830	1.0086	973	236.0083	12.947.1084	234.4717	13.347.5771	0.2321	1.0309	973	236.0083	12.947.1084	234.4717	13.347.5771	0.2321	1.0309			
918	235.8583	12.943.7275	242.5833	13.440.6406	-1.0081	1.0384	974	235.8783	12.911.8662	231.9833	12.769.8096	0.5953	1.0111	974	235.8783	12.911.8662	231.9833	12.769.8096	0.5953	1.0111			
919	235.8933	12.875.5438	231.3383	12.473.3477	0.6931	1.0322	975	235.0533	12.965.8896	239.2467	12.412.9404	-0.4910	1.0445	975	235.0533	12.965.8896	239.2467	12.412.9404	-0.4910	1.0445			
920	235.8633	12.876.1054	245.8567	12.575.9052	-1.5344	1.0240	976	235.7333	12.978.7598	236.0450	12.877.3398	-0.0475	1.0079	976	235.7333	12.978.7598	236.0450	12.877.3398	-0.0475	1.0079			
921	235.8733	13.018.4287	230.8550	12.674.4785	0.7669	1.0271	977	235.6400	12.629.6992	239.4933	13.293.3389	-0.5840	1.0361	977	235.6400	12.629.6992	239.4933	13.293.3389	-0.5840	1.0361			
922	236.0036	12.825.5117	236.8000	12.537.1553	-0.1225	1.0230	978	235.8717	12.907.5928	235.8267	12.885.8193	0.0069	1.0017	978	235.8717	12.907.5928	235.8267	12.885.8193	0.0069	1.0017			
923	235.9817	12.939.0811	234.9283	13.067.7461	0.1592	1.0099	979	235.9800	12.916.2008	239.4082	13.309.1787	-0.5186	1.0304	979	235.9800	12.916.2008	239.4082	13.309.1787	-0.5186	1.0304			
924	235.8207	12.810.7461	234.1150	13.305.3807	0.2589	1.0306	980	236.0467	12.983.2773	243.7017	12.711.0557	-1.1698	1.0214	980	236.0467	12.983.2773	243.7017	12.711.0557	-1.1698	1.0214			
925	235.8917	12.875.9654	235.4283	12.905.5693	0.0707	1.0024	981	235.6083	12.944.7148	230.9183	13.342.6680	0.7086	1.0307	981	235.6083	12.944.7148	230.9183	13.342.6680	0.7086	1.0307			
926	235.9367	12.941.2402	236.8650	12.625.2803	-0.4486	1.0250	982	235.8667	12.901.9131	240.1817	12.883.6533	-0.6582	1.0014	982	235.8667	12.901.9131	240.1817	12.883.6533	-0.6582	1.0014			
927	236.0050	12.964.6038	238.2650	11.991.0503	-0.3489	1.0821	983	236.2450	12.883.2666	235.5383	13.237.7295	0.1071	1.0275	983	236.2450	12.883.2666	235.5383	13.237.7295	0.1071	1.0275			
928	236.8467	12.931.1455	239.6750	13.141.2168	-0.5049	1.0162	984	236.0217	12.895.4775	234.5467	12.521.4102	0.2266	1.0299	984	236.0217	12.895.4775	234.5467	12.521.4102	0.2266	1.0299			
929	236.0067	12.952.4258	237.9783	13.336.5137	-0.2979	1.0297	985	236.0683	12.853.2002	240.1567	12.971.5078	-0.6232	1.0092	985	236.0683	12.853.2002	240.1567	12.971.5078	-0.6232	1.0092			
930	235.6750	12.852.5884	234.0500	12.706.9287	0.2490	1.0115	986	235.8967	12.941.1455	233.7833	13.025.5625	0.3212	1.0065	986	235.8967	12.941.1455	233.7833	13.025.5625	0.3212	1.0065			
931	235.5783	12.957.7254	234.9017	12.864.3408	0.1031	1.0073	987	235.9900	12.872.6025	238.5317	13.215.5898	-0.3855	1.0266	987	235.9900	12.872.6025	238.5317	13.215.5898	-0.3855	1.0266			
932	235.9927	12.921.8965	236.3433	13.015.4209	-0.0527	1.0072	988	235.2433	12.884.0762	239.7883	13.129.6650	-0.6903	1.0191	988	235.2433	12.884.0762	239.7883	13.129.6650	-0.6903	1.0191			
933	235.9800	12.986.0137	236.6217	13.034.8350	0.0544	1.0084	989	236.2217	12.899.0645	244.0833	13.328.0928	-1.1893	1.0341	989	236.2217	12.899.0645	244.0833	13.328.0928	-1.1893	1.0341			
934	236.7865	12.978.4434	231.9950	12.620.8545	0.7338	1.0283	990	235.7800	12.922.9189	241.9150	12.767.5254	-0.9376	1.0122	990	235.7800	12.922.9189	241.9150	12.767.5254	-0.9376	1.0122			
935	235.8288	12.920.9363	234.5583	13.100.0076	0.1928	1.0139	991	236.1100	12.940.8438	237.0117	13.459.5957	-0.1359	1.0401	991	236.1100	12.940.8438	237.0117	13.459.5957	-0.1359	1.0401			
936	235.6567	12.864.8105	236.201.																				

256

ประวัติผู้ เขียน

ชื่อ นามสกุล นางสาวอรรฉรม บุญพะ
วัน เดือน ปีเกิด 13 มีนาคม พ.ศ. 2495
ที่อยู่ 14/3 ถนนศรีปิงเมือง ตำบลช้างคลาน อำเภอเมือง
จังหวัดเชียงใหม่ 50100
ประวัติการศึกษา สำเร็จการศึกษามัธยมศึกษาตอนปลาย โรงเรียนปรินส์รอยแยลล์
วิทยาลัย อำเภอเมือง จังหวัดเชียงใหม่ ปีการศึกษา 2514
สำเร็จการศึกษาระดับอุดมศึกษา สาขาเทคนิคการแพทย์
มหาวิทยาลัยเชียงใหม่ จังหวัดเชียงใหม่ ปีการศึกษา 2518
การทำงาน พ.ศ. 2518 ถึง พ.ศ. 2522 ตำแหน่งนักวิทยาศาสตร์การแพทย์ แผนกชันสูตรโรค
โรงพยาบาลแมคคอร์มิค จังหวัดเชียงใหม่
พ.ศ. 2522 ถึง ปัจจุบัน ตำแหน่งนักวิทยาศาสตร์การแพทย์ ระดับ 6
ภาควิชากุมารเวชศาสตร์ มหาวิทยาลัยเชียงใหม่