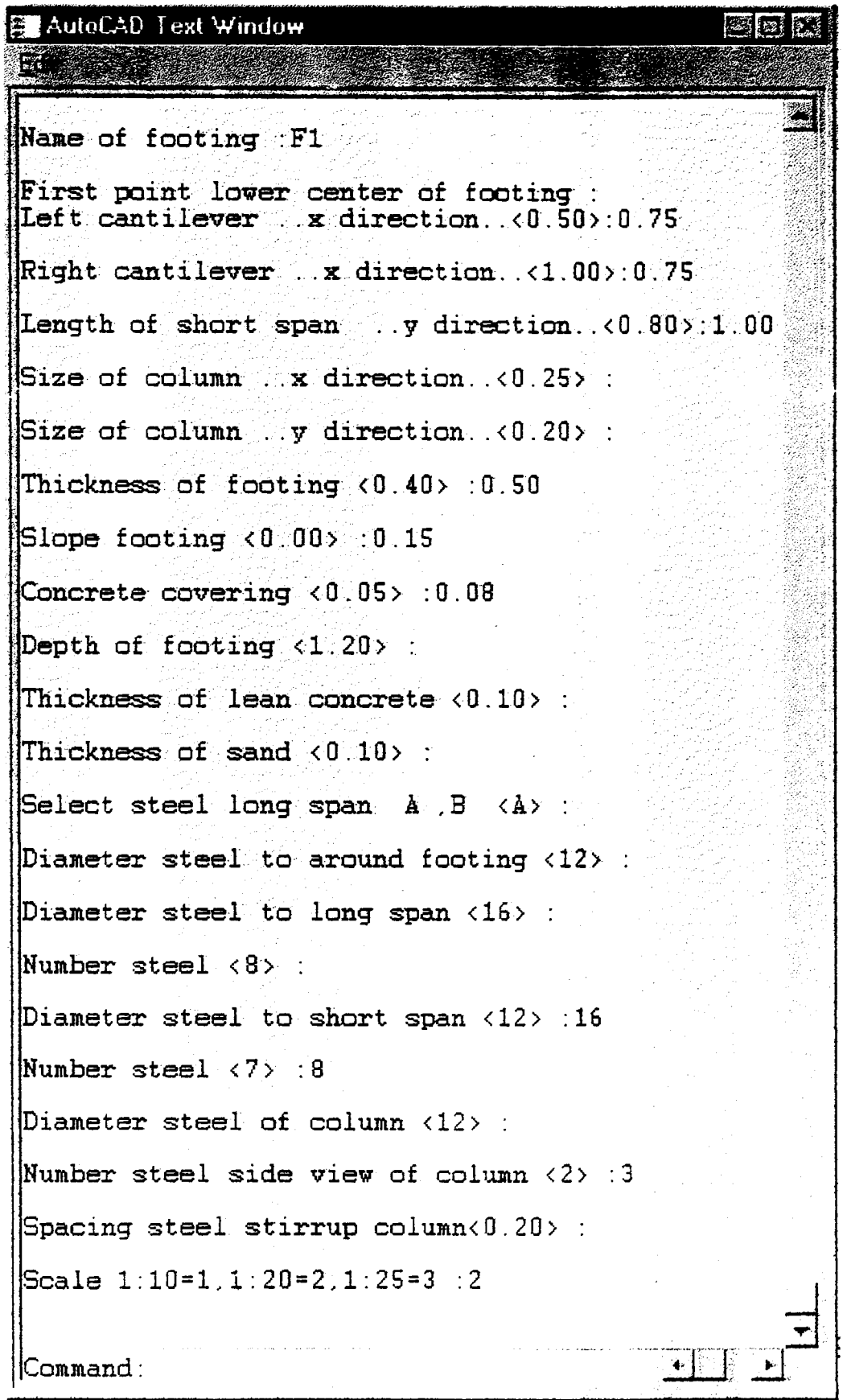
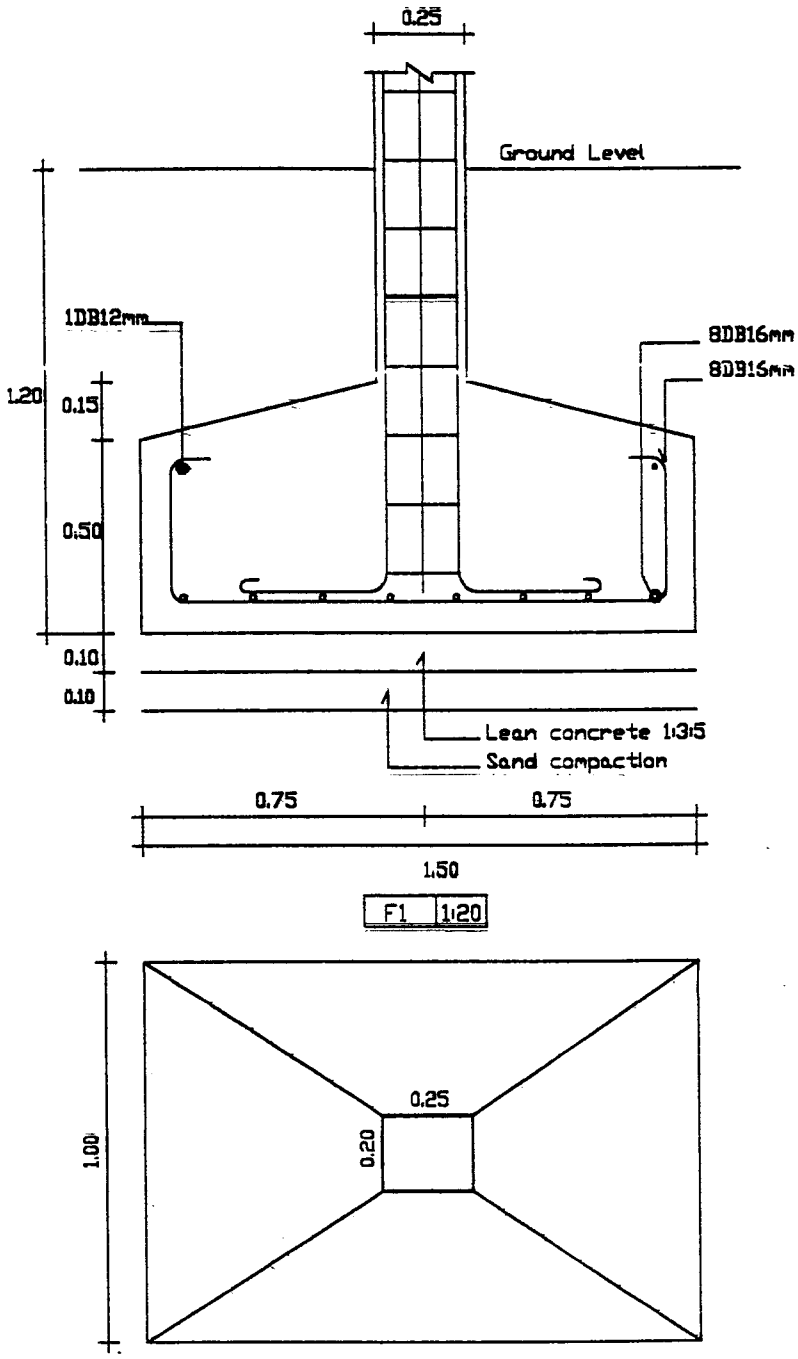
ภาคผนวก

ภาคผนวก ก.

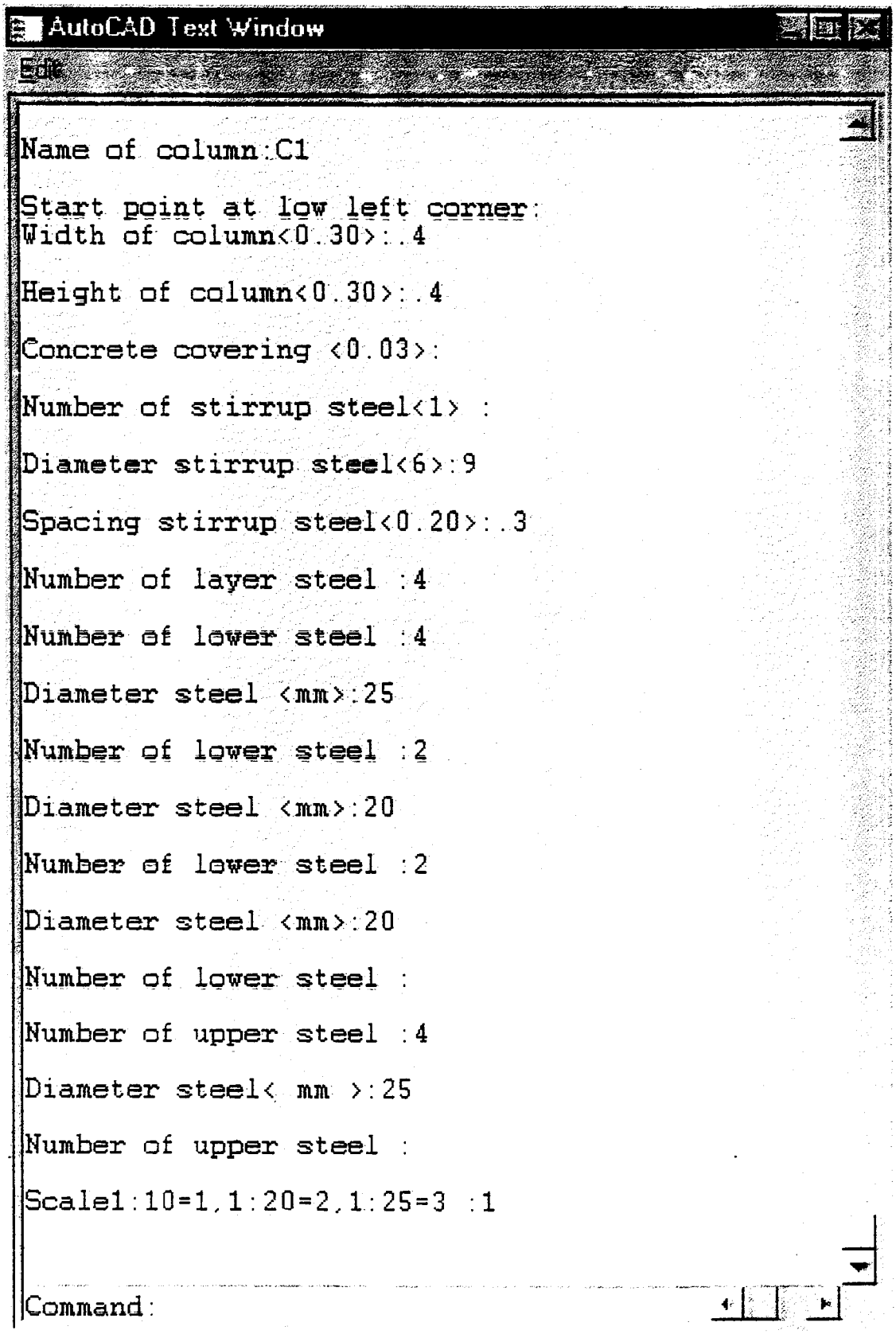
ตัวอย่างทดสอบโปรแกรมเขียนแบบ



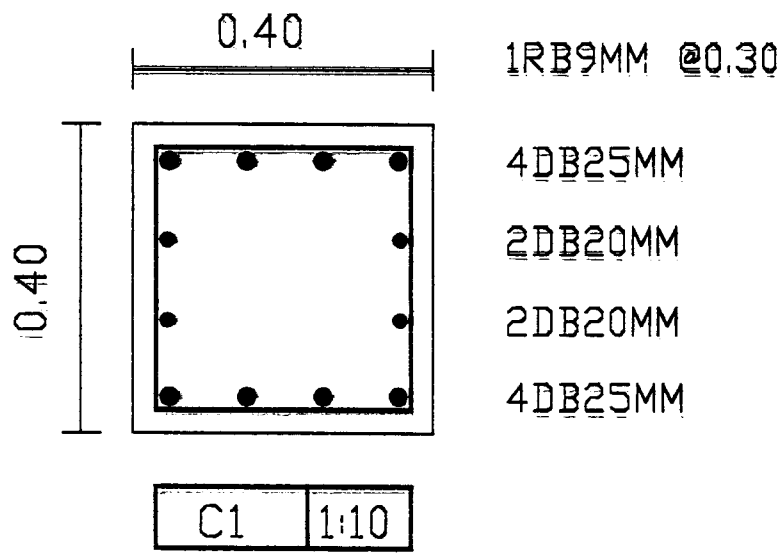
รูปที่ ก.1 ตัวอย่าง การป้อนข้อมูลเพื่อเขียนฐานราก F1



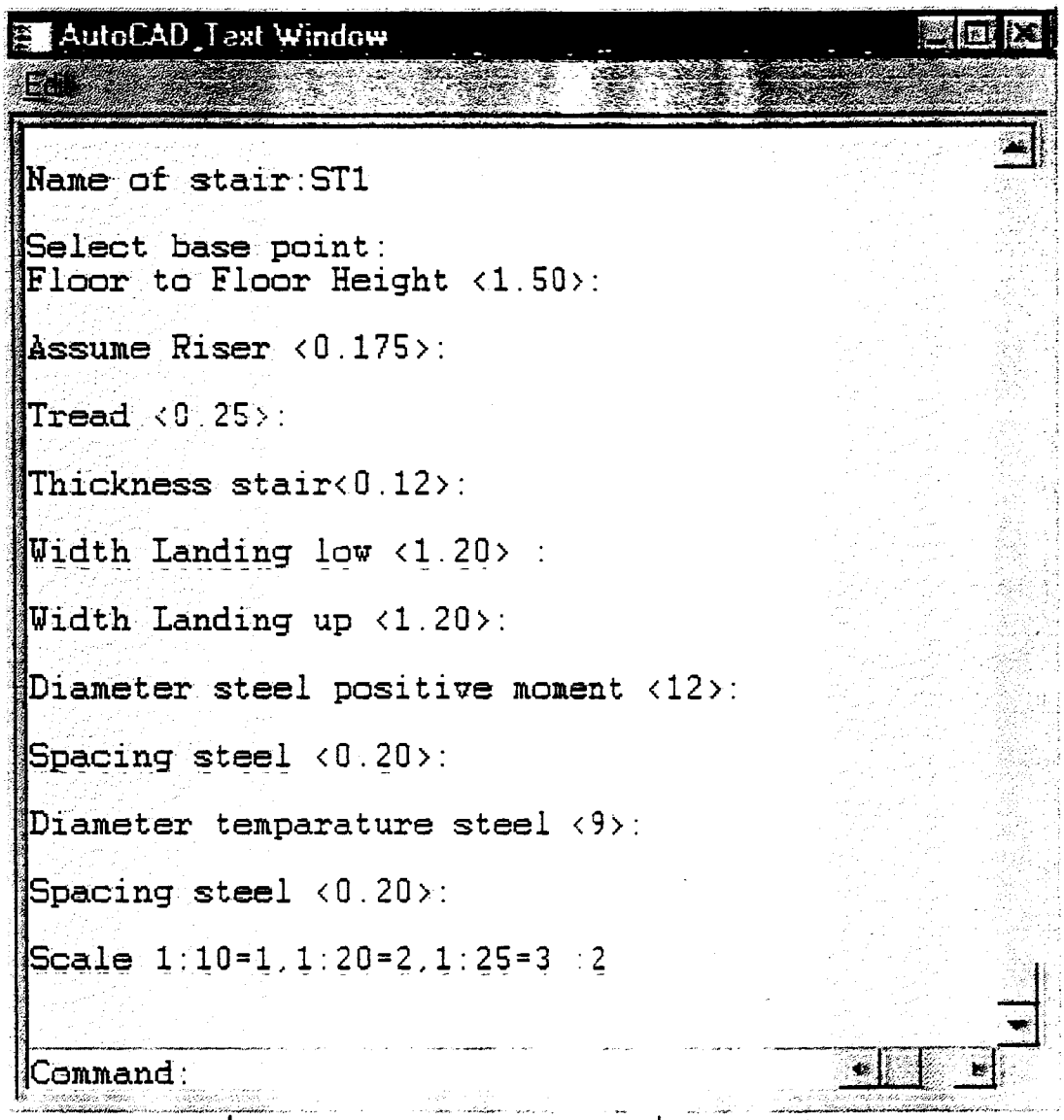
รูป ก.2 แบบขยายฐานราก F1



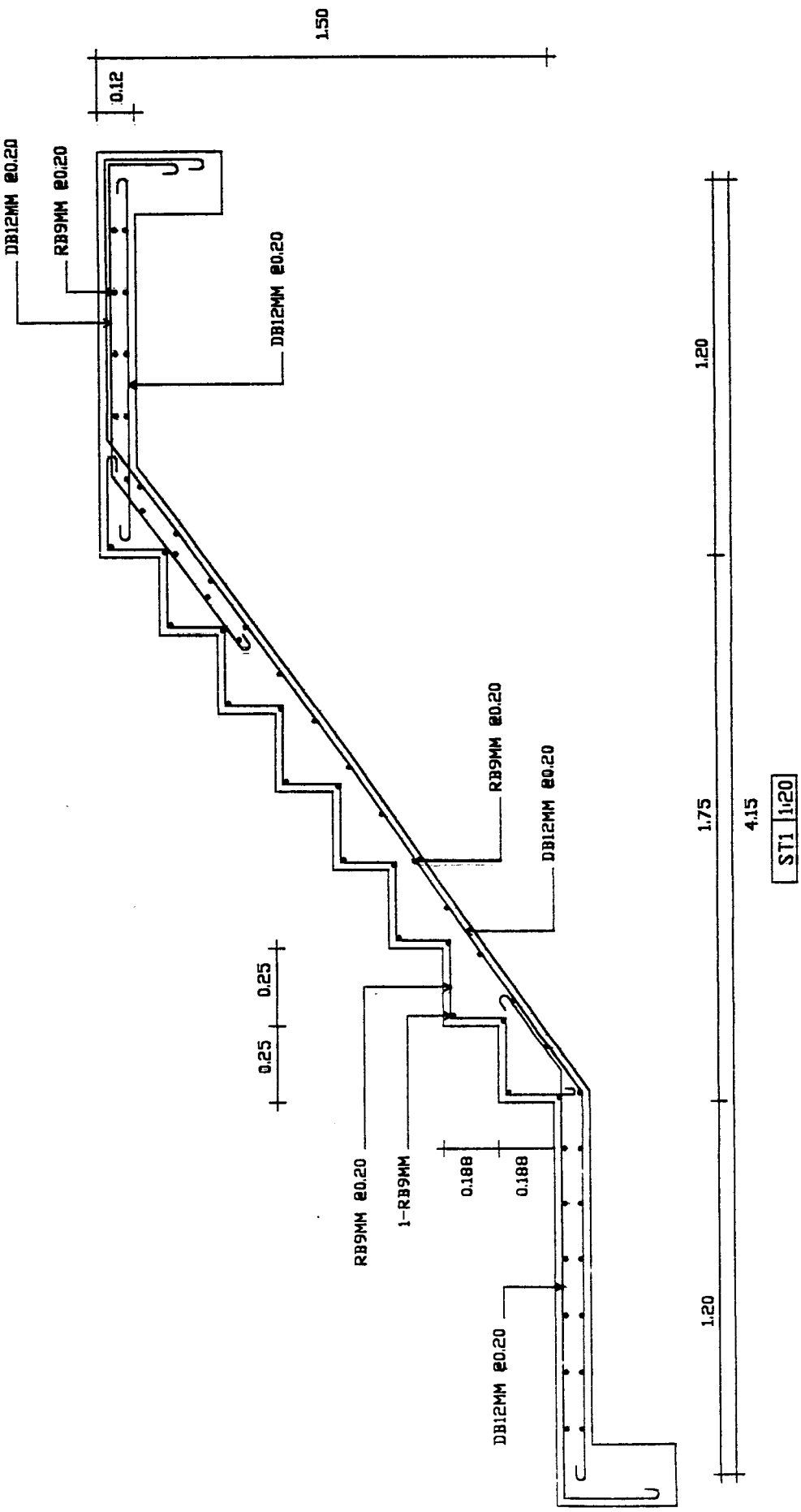
รูปที่ ก.3 ตัวอย่าง การป้อนข้อมูลเพื่อเขียนเสา C1



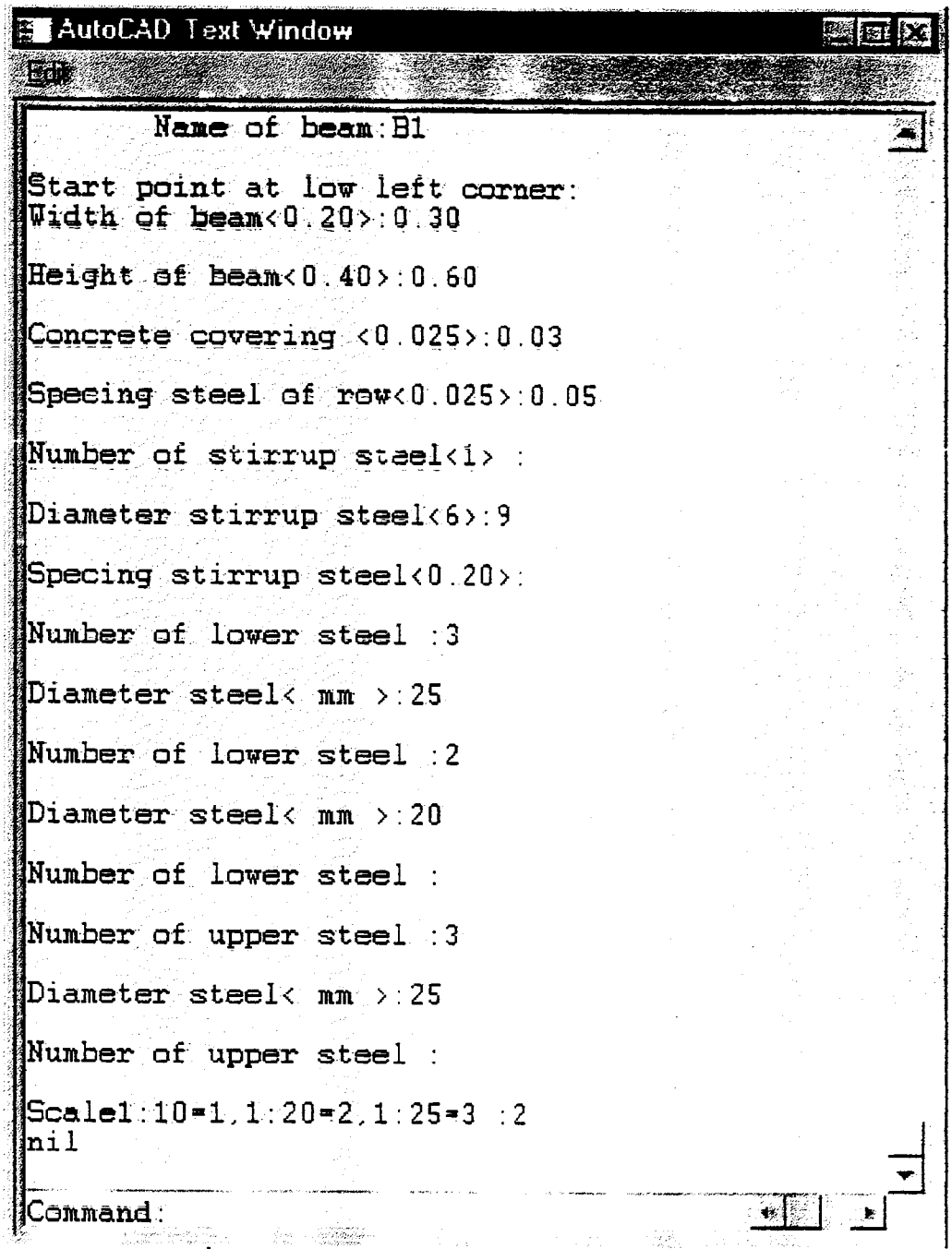
รูป ก.4 แบบขยายเสา C1



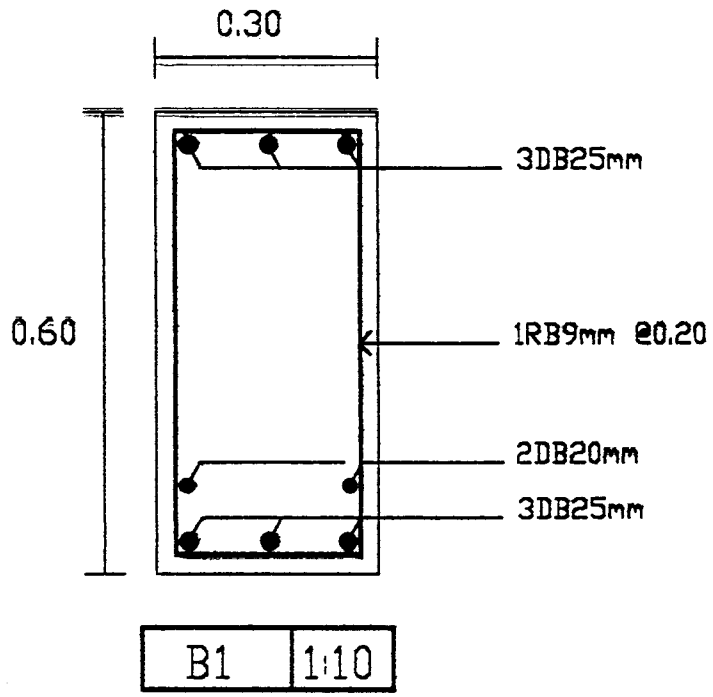
รูปที่ ก.5 ตัวอย่าง การป้อนข้อมูลเพื่อเขียนบันได ST1



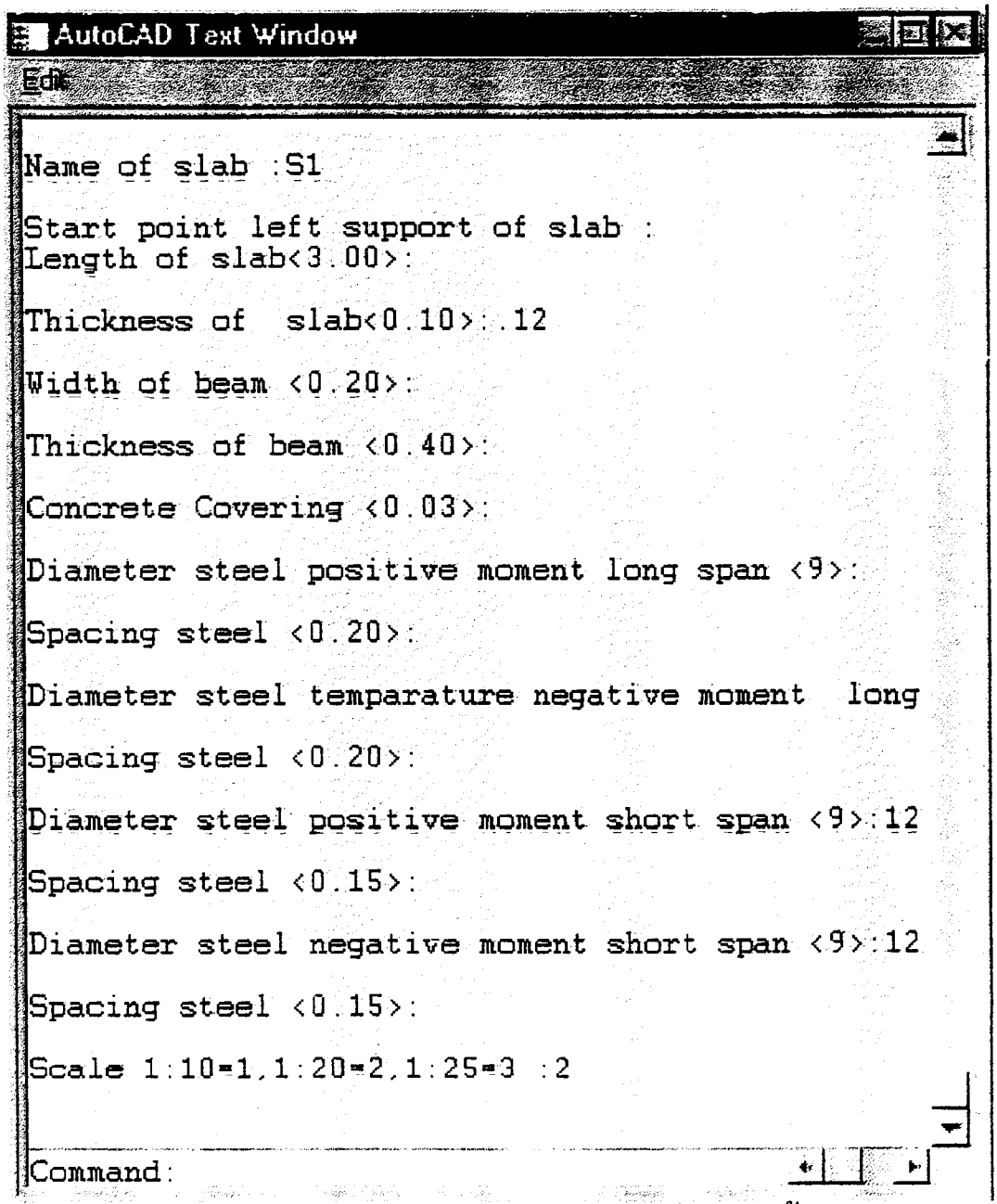
รูป ก.6 แบบขยายบันได ST1



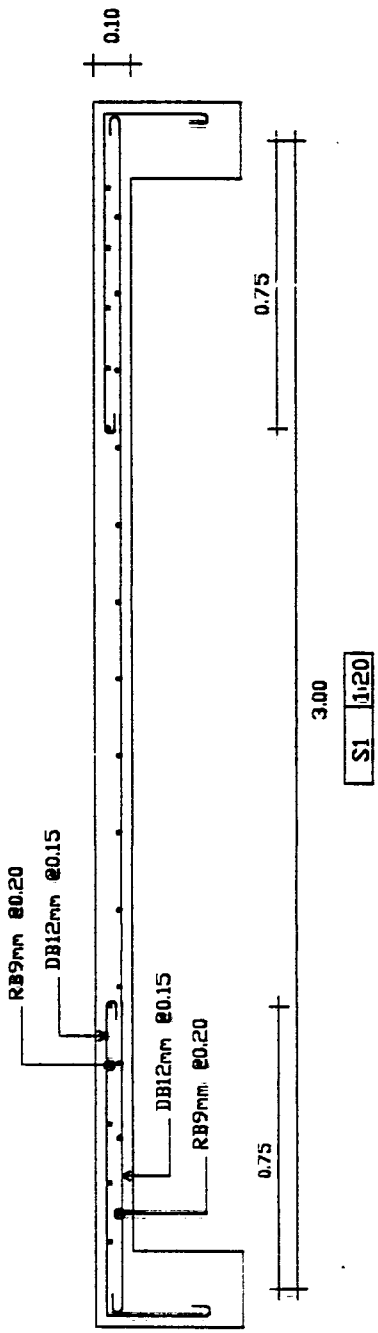
รูปที่ ก.7 ตัวอย่าง การป้อนข้อมูลเพื่อเขียนคาน B1



รูป ก.8 แบบขยายคาน B1



รูปที่ ก.9 ตัวอย่างแสดงการป้อนข้อมูลเพื่อเขียนพื้น S1



รูป ก.10 แบบขยายพื้น S1

ภาคผนวก ข.

รายละเอียดโปรแกรมเชียนแบบ

MENU DRAFTING.MNS

```
//      DEVELOPMENT OF STRUCTURAL DRAFTING SYSTEM USING
MICROCOMPUTER
//      MR. CHUBSORN TASANAKANIJTHAKUL
//      STRUCTURAL ENGINEERING
//      GRADUDTE SCHOOL KHON KAEN UNIVERSITY
//      1998


***menugroup=Chubsorn

***POP1
**draf
[/aDrafting]
[--]
[/MLast Image Menu]^C^C$I=*;
[->/EEnter Filenames]
 [/SSlabs]^C^c^piNSERT;slabs/\
 [/BBeams]^C^c^piNSERT;beams/\
 [/tStairs]^C^c^piNSERT;stairs/\
 [/CColumns]^C^c^piNSERT;columns/\
 [/FFootings]^C^c^piNSERT;footings/\
 [/hShapes]^C^c^piNSERT;shapes/\
 [<-/oConstruction Details]^C^c^pinsert;details/\


[--]
[->/DDesign Structures]
 [/SSlabs]^c^c(load"/r13/drafting/lsp/s12");s12;
 [->/BBeams]
 [/CCross Section beam]^c^c(load"/r13/drafting/lsp/beams");beams;
 [/LLong Section beam]^c^c(load"/r13/drafting/lsp/bs");bs;
```

```
[<-/tCantilever beam]^c^c(load"/r13/drafting/lsp/cb");cb;
[/tStairs]^C^C$I=Chubsorn.STA;$I=*;
[/CColumns]^c^c(load"/r13/drafting/lsp/columns");columns;
[/FFootings]^c^c(load"/r13/drafting/lsp/footings");footings;
[/iFin]^c^c(load"/r13/drafting/lsp/fin2");fin2;
[/fRoof1]^c^c(load"/r13/drafting/lsp/roof4");r1;
[<-/RRoof2]^c^c(load"/r13/drafting/lsp/roof4");r4;
[/eDesign Shapes Steel]^C^C$I=Chubsorn.SHA;$I=*;

[--]
[->/iSlides]
[/SSlabs]^c^cscript;slabs
[/BBeams]^c^cscript;beams
[/CColumns]^c^cscript;columns
[/aSlabs menu]$I=CHUBSORN.CD15;$I=*;
[/eBeams menu]$I=CHUBSORN.CD16;$I=*;
[<-/oColumns menu]$I=CHUBSORN.CD17;$I=*;
[Continue slides]^c^cresume

[--]
[/SSlabs]$I=CHUBSORN.CD10;$I=*;
[/BBeams]$I=CHUBSORN.CD11;$I=*;
[/tStairs]$I=CHUBSORN.CD14;$I=*;
[/CColumns]$I=CHUBSORN.CD12;$I=*;
[/FFootings]^C^C$I=Chubsorn.CD;$I=*;
[/hShapes]$I=CHUBSORN.CD13;$I=*;
[/oConstruction Details]^C^C$I=Chubsorn.CD;$I=*;
[--]


***Image

**STA
```

[Stair Image Menus]

[STAIR(STAIR1U,Stair Up. case1)]^c^c(load"/r13/drafting/lsp/stair1u");su;

[STAIR(STAIR1L,Stair Lo. case1)]^c^c(load"/r13/drafting/lsp/stair1l");stair1l;

[STAIR(STAIR2U,Stair Up. case2)]^c^c(load"/r13/drafting/lsp/stair2u");stair2u;

[STAIR(STAIR2L,Stair Lo. case2)]^c^c(load"/r13/drafting/lsp/stair2l");stair2l;

[STAIR(STAIR3U,Stair Up. case3)]^c^c(load"/r13/drafting/lsp/stair3u");stair3u;

[STAIR(STAIR3L,Stair Lo. case3)]^c^c(load"/r13/drafting/lsp/stair3l");stair3l;

[STAIR(STAIR4U,Stair Up. case4)]^c^c(load"/r13/drafting/lsp/stair4u");stair4u;

[STAIR(STAIR4L,Stair Lo. case4)]^c^c(load"/r13/drafting/lsp/stair4l");stair4l;

[STAIR(STAIR5U,Stair Up. case5)]^c^c(load"/r13/drafting/lsp/stair5u");stair5u;

[STAIR(STAIR5L,Stair Lo. case5)]^c^c(load"/r13/drafting/lsp/stair5l");stair5l;

[STAIR(STAIR6U,Stair Up. case6)]^c^c(load"/r13/drafting/lsp/stair6u");stair6u;

[STAIR(STAIR6L,Stair Lo. case6)]^c^c(load"/r13/drafting/lsp/stair6l");stair6l;

**SHA

[Shape Image Menus]

[SHA(st-01,Shape 1.)]^c^c(load"/r13/drafting/lsp/ang3");a1;

[SHA(st-12,Shape 2.)]^c^c(load"/r13/drafting/lsp/ang3");asc;

[SHA(st-03,Shape 3.)]^c^c(load"/r13/drafting/lsp/ang3");aa2;

[SHA(st-031,Shape 4.)]^c^c(load"/r13/drafting/lsp/ang3");aa2;

[SHA(st-02,Shape 5.)]^c^c(load"/r13/drafting/lsp/ang3");a2;

[SHA(st-022,Shape 6.)]^c^c(load"/r13/drafting/lsp/ang3");a2;

[SHA(st-021,Shape 7.)]^c^c(load"/r13/drafting/lsp/ang3");a2;

[SHA(st-023,Shape 8.)]^c^c(load"/r13/drafting/lsp/ang3");a2;

[SHA(st-05,Shape 13.)]^c^c(load"/r13/drafting/lsp/ang3");a3;

[SHA(st-06,Shape 14.)]^c^c(load"/r13/drafting/lsp/ang3");aaa2;

[SHA(st-061,Shape 15.)]^c^c(load"/r13/drafting/lsp/ang3");aaa2;

[SHA(st-07,Shape 16.)]^c^c(load"/r13/drafting/lsp/ang3");aa4;

[SHA(st-08,Shape 17.)]^c^c(load"/r13/drafting/lsp/ang3");a8;

[SHA(st-09,Shape 18.)]^c^c(load"/r13/drafting/lsp/ang3");aa8;

[SHA(st-10,Shape 19.)]^c^c(load"/r13/drafting/lsp/ang3");a10;

[SHA(st-11,Shape 20.)]^c^c(load"/r13/drafting/lsp/ang3");a9;

[SHA(st-04,Shape 9.)]^c^c(load"/r13/drafting/lsp/ang3");a9;

[SHA(st-041,Shape 10.)]^c^c(load"/r13/drafting/lsp/ang3");a5;

**CD

[Construction Details]

[SLAB(SLAB-01,Slabs)]$I=CHUBSORN.CD10;$I=*;

[BEAM(B-CON1,Beams)]$I=CHUBSORN.CD11;$I=*;

[STAIR(STAIR1U,Stairs)]$I=CHUBSORN.CD14;$I=*;

[COLUMN(C4,Columns)]$I=CHUBSORN.CD12;$I=*;

[SHAPE(STIRMENU,Shapes Steel)]$I=CHUBSORN.CD13;$I=*;

**CD10

[Slabs]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[SLAB(SLAB-01,Short span con.1)]^C^c^piNSERT;SLABS/SLAB-01;

[SLAB(SLAB-02,Long span con.1)]^C^c^piNSERT;SLABS/SLAB-02;

[SLAB(SLAB-03,Short span con.2)]^C^c^piNSERT;SLABS/SLAB-03;

[SLAB(SLAB-04,Long span con.2)]^C^c^piNSERT;SLABS/SLAB-04;

[SLAB(SLAB-09,Short span con.3)]^C^c^piNSERT;SLABS/SLAB-09;

[SLAB(SLAB-10,Long span con.3)]^C^c^piNSERT;SLABS/SLAB-10;

[SLAB(SLAB-06,Short span)]^C^c^piNSERT;SLABS/SLAB-06;

[SLAB(SLAB-08,Long span)]^C^c^piNSERT;SLABS/SLAB-08;

[SLAB(SLAB-05,Joint slab short.)]^C^c^piNSERT;SLABS/SLAB-05;

[SLAB(SLAB-07,Joint slab long.)]^C^c^piNSERT;SLABS/SLAB-07;

[SLAB(SLAB-18,Joint slab)]^C^c^piNSERT;SLABS/SLAB-18;

[SLAB(SLAB-19,Prestress Slab)]^C^c^piNSERT;SLABS/SLAB-19;

[SLAB(SLAB-11,Cantilever Slab 1)]^C^c^piNSERT;SLABS/SLAB-11;

[SLAB(SLAB-12,Cantilever Slab 2)]^C^c^piNSERT;SLABS/SLAB-12;

[SLAB(SLAB-13,Cantilever Slab 3)]^C^c^piNSERT;SLABS/SLAB-13;

[SLAB(SLAB-14,Cantilever Slab 4)]^C^c^piNSERT;SLABS/SLAB-14;

[SLAB(SLAB-15,Ground Slab 1)]^C^c^piNSERT;SLABS/SLAB-15;

[SLAB(SLAB-16,Ground Slab 2)]^C^c^piNSERT;SLABS/SLAB-16;

[SLAB(SLAB-17,Ground Slab 3)]^C^c^piNSERT;SLABS/SLAB-17;

**CD11

[Beams]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[BEAM(B-SIM1,Simple Beam Case1)]^C^c^piNSERT;BEAMS/B-SIM1;

[BEAM(B-SIM2,Simple Beam Case2)]^C^c^piNSERT;BEAMS/B-SIM2;

[BEAM(B-SIM3,Simple Beam Case3)]^C^c^piNSERT;BEAMS/B-SIM3;

[BEAM(B-SIM4,Simple Beam Case4)]^C^c^piNSERT;BEAMS/B-SIM4;

[BEAM(B-SIM5,Simple Beam Case5)]^C^c^piNSERT;BEAMS/B-SIM5;

[BEAM(B-CON1,Continuous Beam Case1)]^C^c^piNSERT;BEAMS/B-CON1;

[BEAM(B-CON2,Continuous Beam Case2)]^C^c^piNSERT;BEAMS/B-CON2;

[BEAM(B-CON3,Continuous Beam Case3)]^C^c^piNSERT;BEAMS/B-CON3;

[BEAM(B-CON4,Continuous Beam Case4)]^C^c^piNSERT;BEAMS/B-CON4;

[BEAM(B-CON5,Continuous Beam Case5)]^C^c^piNSERT;BEAMS/B-CON5;

[BEAM(B-CON6,Continuous Beam Case6)]^C^c^piNSERT;BEAMS/B-CON6;

[BEAM(B-OVER1,Overhanging Beam Case1)]^C^c^piNSERT;BEAMS/B-OVER1;

[BEAM(B-OVER2,Overhanging Beam Case2)]^C^c^piNSERT;BEAMS/B-OVER2;

[BEAM(B-OVER3,Overhanging Beam Case3)]^C^c^piNSERT;BEAMS/B-OVER3;

[BEAM(B-OVER4,Overhanging Beam Case4)]^C^c^piNSERT;BEAMS/B-OVER4;

[BEAM(B-OVER5,Overhanging Beam Case5)]^C^c^piNSERT;BEAMS/B-OVER5;

[BEAM(B-OVER6,Overhanging Beam Case6)]^C^c^piNSERT;BEAMS/B-OVER6;

[BEAM(B-OVER7,Overhanging Beam Case7)]^C^c^piNSERT;BEAMS/B-OVER7;

[BEAM(B-OVER8,Overhanging Beam Case8)]^C^c^piNSERT;BEAMS/B-OVER8;

[BEAM(B-OVER9,Overhanging Beam Case9)]^C^c^piNSERT;BEAMS/B-OVER9;

[BEAM(B-OVER10,Overhanging Beam Case10)]^C^c^piNSERT;BEAMS/B-OVER10;

[BEAM(B-OVER11,Overhanging Beam Case11)]^C^c^piNSERT;BEAMS/B-OVER11;

[BEAM(B-OVER12,Overhanging Beam Case12)]^C^c^piNSERT;BEAMS/B-OVER12;

[BEAM(B-OVER13,Overhanging Beam Case13)]^C^c^piNSERT;BEAMS/B-OVER13;

**CD12

[Columns]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[COLUMN(COL-01,Joint Column Case1)]^C^c^piNSERT;COLUMNS/COL-01;

[COLUMN(COL-02,Joint Column Case2)]^C^c^piNSERT;COLUMNS/COL-02;

[COLUMN(COL-03,Joint Column Case3)]^C^c^piNSERT;COLUMNS/COL-03;

[COLUMN(C4,Column Case1)]^C^c^piNSERT;COLUMNS/C4;

[COLUMN(C6,Column Case2)]^C^c^piNSERT;COLUMNS/C6;

[COLUMN(C6-2,Column Case3)]^C^c^piNSERT;COLUMNS/C6-2;

[COLUMN(C6-3,Column Case4)]^C^c^piNSERT;COLUMNS/C6-3;

[COLUMN(C6-4,Column Case5)]^C^c^piNSERT;COLUMNS/C6-4;

[COLUMN(C6-5,Column Case6)]^C^c^piNSERT;COLUMNS/C6-5;

[COLUMN(C6-6,Column Case7)]^C^c^piNSERT;COLUMNS/C6-6;

[COLUMN(C6-7,Column Case8)]^C^c^piNSERT;COLUMNS/C6-7;

[COLUMN(C8,Column Case9)]^C^c^piNSERT;COLUMNS/C8;

[COLUMN(C8-1,Column Case10)]^C^c^piNSERT;COLUMNS/C8-1;

[COLUMN(C8-2,Column Case11)]^C^c^piNSERT;COLUMNS/C8-2;

[COLUMN(C8-3,Column Case12)]^C^c^piNSERT;COLUMNS/C8-3;

[COLUMN(C8-4,Column Case13)]^C^c^piNSERT;COLUMNS/C8-4;

[COLUMN(C10-1,Column Case14)]^C^c^piNSERT;COLUMNS/C10-1;

[COLUMN(C12-1,Column Case15)]^C^c^piNSERT;COLUMNS/C12-1;

[COLUMN(C12-2,Column Case16)]^C^c^piNSERT;COLUMNS/C12-2;

[COLUMN(C12-3,Column Case17)]^C^c^piNSERT;COLUMNS/C12-3;

[COLUMN(C16-1,Column Case18)]^C^c^piNSERT;COLUMNS/C16-1;

[COLUMN(C16-2,Column Case19)]^C^c^piNSERT;COLUMNS/C16-2;

[COLUMN(C16-3,Column Case20)]^C^c^piNSERT;COLUMNS/C16-3;

[COLUMN(C16-4,Column Case21)]^C^c^piNSERT;COLUMNS/C16-4;

[COLUMN(C16-5,Column Case22)]^C^c^piNSERT;COLUMNS/C16-5;

[COLUMN(C20-1,Column Case23)]^C^c^piNSERT;COLUMNS/C20-1;

**CD13

[Shapes Steel]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[SHAPE(STIR-01,Shape 1)]^C^c^piNSERT;SHAPES/STIR-01;

[SHAPE(STIR-02,Shape 2)]^C^c^piNSERT;SHAPES/STIR-02;

[SHAPE(STIR-021,Shape 3)]^C^c^piNSERT;SHAPES/STIR-021;

[SHAPE(STIR-03,Shape 4)]^C^c^piNSERT;SHAPES/STIR-03;

[SHAPE(STIR-031,Shape 5)]^C^c^piNSERT;SHAPES/STIR-031;

[SHAPE(STIR-04,Shape 6)]^C^c^piNSERT;SHAPES/STIR-04;

[SHAPE(STIR-041,Shape 7)]^C^c^piNSERT;SHAPES/STIR-041;

[SHAPE(STIR-042,Shape 8)]^C^c^piNSERT;SHAPES/STIR-042;

[SHAPE(STIR-043,Shape 9)]^C^c^piNSERT;SHAPES/STIR-043;

[SHAPE(STIR-044,Shape 10)]^C^c^piNSERT;SHAPES/STIR-044;

[SHAPE(STIR-045,Shape 11)]^C^c^piNSERT;SHAPES/STIR-045;

[SHAPE(STIR-046,Shape 12)]^C^c^piNSERT;SHAPES/STIR-046;

[SHAPE(STIR-05,Shape 13)]^C^c^piNSERT;SHAPES/STIR-05;

[SHAPE(STIR-051,Shape 14)]^C^c^piNSERT;SHAPES/STIR-051;

[SHAPE(STIR-052,Shape 15)]^C^c^piNSERT;SHAPES/STIR-052;

[SHAPE(STIR-06,Shape 16)]^C^c^piNSERT;SHAPES/STIR-06;

[SHAPE(STIR-07,Shape 17)]^C^c^piNSERT;SHAPES/STIR-07;

[SHAPE(STIR-08,Shape 18)]^C^c^piNSERT;SHAPES/STIR-08;

[SHAPE(STIR-09,Shape 19)]^C^c^piNSERT;SHAPES/STIR-09;

[SHAPE(STIR-10,Shape 20)]^C^c^piNSERT;SHAPES/STIR-10;

[SHAPE(STIR-11,Shape 21)]^C^c^piNSERT;SHAPES/STIR-11;

[SHAPE(STIR-12,Shape 22)]^C^c^piNSERT;SHAPES/STIR-12;

[SHAPE(STIR-13,Shape 23)]^C^c^piNSERT;SHAPES/STIR-13;

[SHAPE(STIR-14,Shape 24)]^C^c^piNSERT;SHAPES/STIR-14;

[SHAPE(STIR-141,Shape 25)]^C^c^piNSERT;SHAPES/STIR-141;

[SHAPE(STIR-15,Shape 26)]^C^c^piNSERT;SHAPES/STIR-15;

[SHAPE(STIR-16,Shape 27)]^C^c^piNSERT;SHAPES/STIR-16;

[SHAPE(STIR-17,Shape 28)]^C^c^piNSERT;SHAPES/STIR-17;

[SHAPE(STIR-18,Shape 29)]^C^c^piNSERT;SHAPES/STIR-18;

[SHAPE(STIR-19,Shape 30)]^C^c^piNSERT;SHAPES/STIR-19;

[SHAPE(STIR-20,Shape 31)]^C^c^piNSERT;SHAPES/STIR-20;

[SHAPE(STIR-21,Shape 32)]^C^c^piNSERT;SHAPES/STIR-21;

[SHAPE(STIR-22,Shape 33)]^C^c^piNSERT;SHAPES/STIR-22;

[SHAPE(STIR-221,Shape 34)]^C^c^piNSERT;SHAPES/STIR-221;

[SHAPE(STIR-222,Shape 35)]^C^c^piNSERT;SHAPES/STIR-222;

[SHAPE(STIR-223,Shape 36)]^C^c^piNSERT;SHAPES/STIR-223;

[SHAPE(STIR-224,Shape 37)]^C^c^piNSERT;SHAPES/STIR-224;

[SHAPE(STIR-23,Shape 38)]^C^c^piNSERT;SHAPES/STIR-23;

[SHAPE(STIR-231,Shape 39)]^C^c^piNSERT;SHAPES/STIR-231;

[SHAPE(STIR-232,Shape 40)]^C^c^piNSERT;SHAPES/STIR-232;

[SHAPE(STIR-24,Shape 41)]^C^c^piNSERT;SHAPES/STIR-24;

[SHAPE(STIR-241,Shape 42)]^C^c^piNSERT;SHAPES/STIR-241;

[SHAPE(STIR-242,Shape 43)]^C^c^piNSERT;SHAPES/STIR-242;

[SHAPE(STIR-25,Shape 44)]^C^c^piNSERT;SHAPES/STIR-25;

[SHAPE(STIR-26,Shape 45)]^C^c^piNSERT;SHAPES/STIR-26;

[SHAPE(STIR-27,Shape 46)]^C^c^piNSERT;SHAPES/STIR-27;

[SHAPE(STIR-28,Shape 47)]^C^c^piNSERT;SHAPES/STIR-28;

[SHAPE(STIR-281,Shape 48)]^C^c^piNSERT;SHAPES/STIR-281;

[SHAPE(STIR-29,Shape 49)]^C^c^piNSERT;SHAPES/STIR-29;

[SHAPE(STIR-30,Shape 50)]^C^c^piNSERT;SHAPES/STIR-30;

[SHAPE(STIR-31,Shape 51)]^C^c^piNSERT;SHAPES/STIR-31;

[SHAPE(STIR-32,Shape 52)]^C^c^piNSERT;SHAPES/STIR-32;

[SHAPE(STIR-33,Shape 53 Dia. 25mm)]^C^c^piNSERT;SHAPES/STIR-33;

[SHAPE(STIR-34,Shape 54 Dia. 20mm)]^C^c^piNSERT;SHAPES/STIR-34;

[SHAPE(STIR-35,Shape 55 Dia. 16mm)]^C^c^piNSERT;SHAPES/STIR-35;

[SHAPE(STIR-36,Shape 56 Dia. 12mm)]^C^c^piNSERT;SHAPES/STIR-36;

[SHAPE(STIR-37,Shape 57 Dia. 9mm)]^C^c^piNSERT;SHAPES/STIR-37;

[SHAPE(STIR-38,Shape 58 Dia. 6mm)]^C^c^piNSERT;SHAPES/STIR-38;

**CD14

[Stairs]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[STAIR(STAIR1U,Stair Up.case1)]^C^c^piNSERT;STAIRS/STAIR1U;

[STAIR(STAIR1L,Stair Lo.case1)]^C^c^piNSERT;STAIRS/STAIR1L;
[STAIR(STAIR2U,Stair Up.case2)]^C^c^piNSERT;STAIRS/STAIR2U;
[STAIR(STAIR2L,Stair Lo.case2)]^C^c^piNSERT;STAIRS/STAIR2L;
[STAIR(STAIR3U,Stair Up.case3)]^C^c^piNSERT;STAIRS/STAIR3U;
[STAIR(STAIR3L,Stair Lo.case3)]^C^c^piNSERT;STAIRS/STAIR3L;
[STAIR(STAIR4U,Stair Up.case4)]^C^c^piNSERT;STAIRS/STAIR4U;
[STAIR(STAIR4L,Stair Lo.case4)]^C^c^piNSERT;STAIRS/STAIR4L;
[STAIR(STAIR5U,Stair Up.case5)]^C^c^piNSERT;STAIRS/STAIR5U;
[STAIR(STAIR5L,Stair Lo.case5)]^C^c^piNSERT;STAIRS/STAIR5L;
[STAIR(STAIR6U,Stair Up.case6)]^C^c^piNSERT;STAIRS/STAIR6U;
[STAIR(STAIR6L,Stair Lo.case6)]^C^c^piNSERT;STAIRS/STAIR6L;

**CD15
[Slabs]
[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;
[SLAB(SLAB-01,Short span con.1)]^C^c^pVSLIDE;SLABS/SLAB-01;
[SLAB(SLAB-02,Long span con.1)]^C^c^pVSLIDE;SLABS/SLAB-02;
[SLAB(SLAB-03,Short span con.2)]^C^c^pVSLIDE;SLABS/SLAB-03;
[SLAB(SLAB-04,Long span con.2)]^C^c^pVSLIDE,SLABS/SLAB-04;
[SLAB(SLAB-09,Short span con.3)]^C^c^pVSLIDE;SLABS/SLAB-09;
[SLAB(SLAB-10,Long span con.3)]^C^c^pVSLIDE;SLABS/SLAB-10;
[SLAB(SLAB-06,Short span)]^C^c^pVSLIDE;SLABS/SLAB-06;
[SLAB(SLAB-08,Long span)]^C^c^pVSLIDE;SLABS/SLAB-08;
[SLAB(SLAB-05,Joint slab short.)]^C^c^pVSLIDE;SLABS/SLAB-05;
[SLAB(SLAB-07,Joint slab long.)]^C^c^pVSLIDE;SLABS/SLAB-07;
[SLAB(SLAB-18,Joint slab)]^C^c^pVSLIDE;SLABS/SLAB-18;
[SLAB(SLAB-19,Prestress Slab)]^C^c^pVSLIDE;SLABS/SLAB-19;
[SLAB(SLAB-11,Cantilever Slab 1)]^C^c^pVSLIDE;SLABS/SLAB-11;
[SLAB(SLAB-12,Cantilever Slab 2)]^C^c^pVSLIDE;SLABS/SLAB-12;
[SLAB(SLAB-13,Cantilever Slab 3)]^C^c^pVSLIDE;SLABS/SLAB-13;
[SLAB(SLAB-14,Cantilever Slab 4)]^C^c^pVSLIDE;SLABS/SLAB-14;
[SLAB(SLAB-15,Ground Slab 1)]^C^c^pVSLIDE;SLABS/SLAB-15;

[SLAB(SLAB-16,Ground Slab 2)]^C^c^pVSLIDE;SLABS/SLAB-16;
[SLAB(SLAB-17,Ground Slab 3)]^C^c^pVSLIDE;SLABS/SLAB-17;

**CD16
[Beams]
[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;
[BEAM(B-SIM1,Simple Beam Case1)]^C^c^pVSLIDE;BEAMS/B-SIM1;
[BEAM(B-SIM2,Simple Beam Case2)]^C^c^pVSLIDE;BEAMS/B-SIM2;
[BEAM(B-SIM3,Simple Beam Case3)]^C^c^pVSLIDE;BEAMS/B-SIM3;
[BEAM(B-SIM4,Simple Beam Case4)]^C^c^pVSLIDE;BEAMS/B-SIM4;
[BEAM(B-SIM5,Simple Beam Case5)]^C^c^pVSLIDE;BEAMS/B-SIM5;
[BEAM(B-CON1,Continuous Beam Case1)]^C^c^pVSLIDE;BEAMS/B-CON1;
[BEAM(B-CON2,Continuous Beam Case2)]^C^c^pVSLIDE;BEAMS/B-CON2;
[BEAM(B-CON3,Continuous Beam Case3)]^C^c^pVSLIDE;BEAMS/B-CON3;
[BEAM(B-CON4,Continuous Beam Case4)]^C^c^pVSLIDE;BEAMS/B-CON4;
[BEAM(B-CON5,Continuous Beam Case5)]^C^c^pVSLIDE;BEAMS/B-CON5;
[BEAM(B-CON6,Continuous Beam Case6)]^C^c^pVSLIDE;BEAMS/B-CON6;
[BEAM(B-OVER1,Overhanging Beam Case1)]^C^c^pVSLIDE;BEAMS/B-OVER1;
[BEAM(B-OVER2,Overhanging Beam Case2)]^C^c^pVSLIDE;BEAMS/B-OVER2;
[BEAM(B-OVER3,Overhanging Beam Case3)]^C^c^pVSLIDE;BEAMS/B-OVER3;
[BEAM(B-OVER4,Overhanging Beam Case4)]^C^c^pVSLIDE;BEAMS/B-OVER4;
[BEAM(B-OVER5,Overhanging Beam Case5)]^C^c^pVSLIDE;BEAMS/B-OVER5;
[BEAM(B-OVER6,Overhanging Beam Case6)]^C^c^pVSLIDE;BEAMS/B-OVER6;
[BEAM(B-OVER7,Overhanging Beam Case7)]^C^c^pVSLIDE;BEAMS/B-OVER7;
[BEAM(B-OVER8,Overhanging Beam Case8)]^C^c^pVSLIDE;BEAMS/B-OVER8;
[BEAM(B-OVER9,Overhanging Beam Case9)]^C^c^pVSLIDE;BEAMS/B-OVER9;
[BEAM(B-OVER10,Overhanging Beam Case10)]^C^c^pVSLIDE;BEAMS/B-OVER10;
[BEAM(B-OVER11,Overhanging Beam Case11)]^C^c^pVSLIDE;BEAMS/B-OVER11;
[BEAM(B-OVER12,Overhanging Beam Case12)]^C^c^pVSLIDE;BEAMS/B-OVER12;

[BEAM(B-OVER13,Overhanging Beam Case13)]^C^c^pVSLIDE;BEAMS/B-OVER13;

**CD17

[Columns]

[DETAILS(CONSTDET,Details Main Menu)]$I=CHUBSORN.CD;$I=*;

[COLUMN(COL-01,Joint Column Case1)]^C^c^pVSLIDE;COLUMNS/COL-01;

[COLUMN(COL-02,Joint Column Case2)]^C^c^pVSLIDE;COLUMNS/COL-02;

[COLUMN(COL-03,Joint Column Case3)]^C^c^pVSLIDE;COLUMNS/COL-03;

[COLUMN(C4,Column Case1)]^C^c^pVSLIDE;COLUMNS/C4;

[COLUMN(C6,Column Case2)]^C^c^pVSLIDE;COLUMNS/C6;

[COLUMN(C6-2,Column Case3)]^C^c^pVSLIDE;COLUMNS/C6-2;

[COLUMN(C6-3,Column Case4)]^C^c^pVSLIDE;COLUMNS/C6-3;

[COLUMN(C6-4,Column Case5)]^C^c^pVSLIDE;COLUMNS/C6-4;

[COLUMN(C6-5,Column Case6)]^C^c^pVSLIDE;COLUMNS/C6-5;

[COLUMN(C6-6,Column Case7)]^C^c^pVSLIDE;COLUMNS/C6-6;

[COLUMN(C6-7,Column Case8)]^C^c^pVSLIDE;COLUMNS/C6-7;

[COLUMN(C8,Column Case9)]^C^c^pVSLIDE;COLUMNS/C8;

[COLUMN(C8-1,Column Case10)]^C^c^pVSLIDE;COLUMNS/C8-1;

[COLUMN(C8-2,Column Case11)]^C^c^pVSLIDE;COLUMNS/C8-2;

[COLUMN(C8-3,Column Case12)]^C^c^pVSLIDE;COLUMNS/C8-3;

[COLUMN(C8-4,Column Case13)]^C^c^pVSLIDE;COLUMNS/C8-4;

[COLUMN(C10-1,Column Case14)]^C^c^pVSLIDE;COLUMNS/C10-1;

[COLUMN(C12-1,Column Case15)]^C^c^pVSLIDE;COLUMNS/C12-1;

[COLUMN(C12-2,Column Case16)]^C^c^pVSLIDE;COLUMNS/C12-2;

[COLUMN(C12-3,Column Case17)]^C^c^pVSLIDE;COLUMNS/C12-3;

[COLUMN(C16-1,Column Case18)]^C^c^pVSLIDE;COLUMNS/C16-1;

[COLUMN(C16-2,Column Case19)]^C^c^pVSLIDE;COLUMNS/C16-2;

[COLUMN(C16-3,Column Case20)]^C^c^pVSLIDE;COLUMNS/C16-3;

[COLUMN(C16-4,Column Case21)]^C^c^pVSLIDE;COLUMNS/C16-4;

[COLUMN(C16-5,Column Case22)]^C^c^pVSLIDE;COLUMNS/C16-5;

[COLUMN(C20-1,Column Case23)]^C^c^pVSLIDE;COLUMNS/C20-1;

ACAD.LSP

```
; Load the Drafting menu
(defun DRAFTING ()
(command "_menuunload" "Chubsom")
(command "_menuload" "DRAFTING.MNS")
(menucmd "P7=+Chubsom.pop1")
(princ)
)
(if S::STARTUP
(setq s::startup (append s::startup '((drafting)) )) (defun s::startup () (drafting) )
)
```

ACAD.INI

```
[General]
Measure=0
PrototypeDwg=acad.dwg
ACAD=D:\r13\com\SUPPORT;D:\r13\win\SUPPORT;D:\r13\win\TUTORIAL;D:\r13\
com\FONTS;D:\r13\DRAFTING
ACADHELP=D:\r13\win\SUPPORT\HELP
ACADDRV=D:\r13\win\DRV
AVEPAGEDIR=D:\r13\win
ACADPAGEDIR=D:\r13\win
AVECFG=D:\r13\win
AVEFACEDIR=D:\r13\win
ACADLOGFILE=D:\r13\win\ACAD.LOG
ACADMAXMEM=4000000

[Prototype Drawing]
acad.dwg=Standard Imperial
acadiso.dwg=Metric/ISO Size A3
us_arch.dwg=U.S. Architectural
us_mech.dwg=U.S. Mechanical

[MTE Fonts]
txt=Arial;400;0
complex=Times New Roman;400;0
gothice=Times New Roman;400;0
gothicg=Times New Roman;400;0
gothici=Times New Roman;400;1
greekc=Symbol;400;0
greeks=Symbol;400;0
italic=Times New Roman;400;1
italicc=Times New Roman;400;1
italict=Times New Roman;700;1
```

SLABS.LSP

```
(vmon)
(defun c:slabs(/ pt1 pt2 pt3 pt4 pt5 pt6 pt7 pt8 pt9 pt10 px1 px2 px3 px4  px5
px6 px7 px8 px9 py1 py2 py3 py4)
                (setvar "blipmode" 0)
                (setvar "cmdecho" 0)
                        (setq tb1 (getstring "\nName of slab :"))
                        (setq bm (getpoint "\nStart point left support of slab
:"))
                        (setq l1 (getdist bm "\nLength of slab<3.00>:"))
                    (if (not l1)
                        (setq l1 3.00)
                    )
                (setq ts1 (getdist bm "\nThickness of  slab<0.10>:"))
                    (if (not ts1)
                        (setq ts1 0.10)
                    )
                        (setq wb (getreal "\nWidth of beam <0.20>:"))
                 (if (not wb)
                        (setq wb 0.20)
                    )
                        (setq tk1 (getreal "\nThickness of beam <0.40>:"))
                    (if (not tk1)
                        (setq tk1 0.40)
                    )
                        (setq co (getreal "\nConcrete Covering <0.03>:"))
                    (if (not co)
                        (setq co 0.03)
                    )
                    (if (>= co (/ ts1 2))
                       (setq co 0.03)
                    )
                (setq bm1 (polar bm 0 l1))
```

```
(setq pt1 (polar bm 0 (/ wb 2)))
(setq pt2 (polar bm 0 (- l1 (/ wb 2))))
(setq pt3 (polar pt2 (/ pi -2) (- tk1 ts1)))
(setq pt4 (polar pt3 0 wb))
(setq pt5 (polar pt4 (/ pi 2) tk1))
(setq pt6 (polar pt5 pi (+ l1 wb)))
(setq pt7 (polar pt6 (/ pi -2) tk1))
(setq pt8 (polar pt7 0 wb))
(setq pt9 (polar pt6 (/ pi 1.3) 0.50))
(setq pt10 (polar pt4 (/ pi -4) 0.50))
(command "pline" pt1 pt2 pt3 pt4 pt5 pt6 pt7 pt8 pt1 "")
(setq px1 (polar bm (/ pi -2)(+ 0.15 (- tk1 ts1))))
(setq px2 (polar bm1 (/ pi -2)(+ 0.15 (- tk1 ts1))))
(command "pline" (polar px1 pi 0.025) px1 px2 (polar px2 0 0.025)"")
(command "pline" (polar px1 (/ pi 2) 0.075) (polar px1 (/ pi -2) 0.025)"")
(command "pline" (polar px2 (/ pi 2) 0.075) (polar px2 (/ pi -2) 0.025)"")

(setq py1 (polar pt5 0 0.10))
(setq py2 (polar py1 (/ pi -2) ts1))
(command "pline" (polar py1 (/ pi 2) 0.025) py1 py2 (polar py2 (/ pi -2)
0.025)"")
(command "pline" (polar py1 pi 0.03) (polar py1 0 0.025)"")
(command "pline" (polar py2 pi 0.03) (polar py2 0 0.025)"")

(setq px3 (polar px1 0 (/ l1 2)))
(setq px4 (polar px3 (/ pi -2) 0.08))
(setq l1 (rtos l1 2 2))
(command "text" px4 0.035 0.00 l1)
(setq l1 (atof l1))

(setq l2 (/ l1 4))
(setq px5 (polar px1 (/ pi 2) 0.05))
(setq px6 (polar px5 0 l2))
```

```
        (setq px7 (polar px5 0 (/ l2 2.5)))
        (setq px8 (polar px7 (/ pi 2) 0.025))
        (setq px9 (polar px8 0 (- l1 l2)))
        (setq l2 (rtos l2 2 2))
(command "pline" (polar px5 pi 0.025) (polar px6 0 0.025)"")
(command "pline" (polar px6 (/ pi 2) 0.025) (polar px6 (/ pi -2) 0.025)"")
(command "text" px8 0.03 0.00 l2)
(command "text"  px9 0.035 0.00 l2)
        (setq l2 (atof l2))

        (setq py3 (polar py1 (/ pi -2) (/ ts1 1.5)))
        (setq py4 (polar py3 0 0.06))
        (setq ts1 (rtos ts1 2 2))
(command "text" py4 0.035 0.00 ts1)
        (setq ts1 (atof ts1))

        (setq pd1 px4)
        (setq pd2 (polar pd1 (/ pi -2) 0.05))
        (setq pd3 (polar pd2 pi 0.17))
        (setq pd4 (polar pd3 (/ pi -2) 0.08))
        (setq pd5 (polar pd4 0 0.20))
        (setq pd6 (polar pd5 0 0.14))
        (setq pd7 (polar pd6 (/ pi 2) 0.08))
        (setq pd8 (polar pd7 pi 0.14))
        (setq pd9 (polar pd4 0 0.06))
        (setq pd10 (polar pd9 (/ pi 2) 0.015))
        (setq pd11 (polar pd5 0 0.02))
        (setq pd12 (polar pd11 (/ pi 2) 0.015))
 (command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5 "")
 (command "text" pd10 0.04 0.00 tb1)

 (command "zoom" "w" pt9 pt10 )
```

```
(command "mirror"  px7 (polar px6 (/ pi 2) 0.025) "" px3 (polar px3 (/ pi 2) 0.50) "")
1.          (command "_pedit" pd2 "w" 0.005 "")

                (setq dia (getreal "\nDiameter steel positive moment long span <mm>:"))
                (if (< dia 12.0)(setq ch1 "RB")(setq ch1 "DB"))
                (setq sps (getreal "\nSpacing steel <0.20>:"))
                (if (not sps)
                (setq sps 0.20)
                )
                (setq dial (getreal "\nDiameter steel temparature negative moment  long span <mm>:"))
                (if (< dial 12.0)(setq ch2 "RB")(setq ch2 "DB"))

                (setq spl (getreal "\nSpacing steel <0.20>:"))
                (if (not spl)
                (setq spl 0.20)
                )

        (setq dia (/ dia 1000))
        (setq ra (/ dia 2))
        (setq ps1 (polar bm (/ pi 2) co))
        (setq psl (polar ps1 pi (/ wb 4)))
        (setq ps2 (polar psl (/ pi 2) 0.025))
        (setq ps3 (polar bm1 (/ pi 2) co))
        (setq psr (polar ps3 0 (/ wb 4)))
        (setq ps4 (polar psr (/ pi 2) 0.025))
        (setq ps5 (polar ps1 0 (+ (/ wb 2) 0.10)))
        (setq ps6 (polar ps5 (/ pi 2) (+ ra 0.0028)))
        (setq x1 (- l1 (+ wb 0.20)))
        (setq sp sps)
        (setq ns (/ x1 sp))
```

```
            (if (= ns (fix ns))
                (setq ro ns)
                (setq ro (+ (fix ns) 1))
            )
                (setq sp (/ x1 ro))
        (setq i 1)
        (while (<= i ro)
            (command "circle" ps6 ra)
                (setq ps6 (polar ps6 0 sp))
                (setq i (+ i 1))
        )
            (command "circle" ps6 ra)

                (setq pss ps5)
                (setq ps7 (polar pss (/ pi -2) (+ co 0.20)))
                (setq ps8 (polar ps7 0 0.15))
                (command "circle" (polar ps5 (/ pi 2) (+ ra 0.0028)) (* dia 1.5))
            (command "pline" ps8 ps7 (polar (polar ps5 (/ pi 2) (+ ra 0.0028)) (/ pi
-2) (* dia 1.5)) "")

                (setq dia (* dia 1000))
                (setq dia (rtos dia 2 0))
                (setq sps (rtos sps 2 2))
                (setq txo (strcat (strcat ch1 dia "mm @") sps""))
            (command "text" (polar ps8 0 0.025) 0.035 0.0 txo)
                (setq dia (atoi dia))
                (setq dia (/ dia 1000))
                (setq sps (atof sps))
            (command "pline"  (polar ps4 pi 0.035) ps4 "a"  psr "l" psl "a" ps2 "l"
(polar ps2 0 0.035)"")
            (command "_pedit" (polar ps5 pi 0.05)  "w" 0.005 "")

                (setq pu1 (polar pt6 (/ pi -4) (* 1.414 co)))
```

```
                (setq pu2 (polar pu1 (/ pi -2) (/ (* tk1 2) 3)))
                (setq pu3 (polar pu2 0 0.025))
                (setq pu4 (polar pu1 0 (+ (/ l1 4) (- (/ wb 2) co))))
                (setq pu5 (polar pu4 (/ pi -2) 0.025))
          (command "pline" (polar pu3 (/ pi 2) 0.035) pu3 "a" pu2 "l" pu1 pu4 "a"
pu5 "l" (polar pu5 pi 0.035)"")
          (command "_pedit" pu1  "w" 0.005 "")

                (setq dial (/ dial 1000))
                (setq ral (/ dial 2))
                (setq pzz (polar pu1 (/ pi -2) (+ ral 0.0028)))
                (setq pz (polar pzz 0 (- (+ wb 0.025) co)))
                (setq x2 (- (- l2 0.025) (/ wb 2)))
                (setq ns1 (/ x2 spl))
           (if (= ns1 (fix ns1))
                (setq rol ns1)
                (setq rol (+ (fix ns1) 1))
           )
                (setq sp1 (/ x2 rol))
     (setq i 1)
     (while (<= i rol)
          (command "circle" pz ral)
               (setq pz (polar pz 0 sp1))
               (setq i (+ i 1))
     )
          (command "circle" pz ral)
          (command "mirror"  pu1 "" px3 (polar px3 (/ pi 2) 0.50) "")
          (command "mirror"  "auto" (polar pu1 (/ pi 2) (+ co 0.05)) (polar pu5 0
0.05) "" px3 (polar px3 (/ pi 2) 0.50) "")

                (setq psa (polar ps5 0 (/ sp 2)))
                (setq ps11 (polar psa (/ pi -2) (+ co 0.10)))
                (setq ps12 (polar ps11 0 0.15))
```

```
(command "pline" ps12 ps11 psa  (polar psa (/ pi -1.5) 0.025) psa
(polar psa (/ pi -3) 0.025)"")

(setq ps (polar pu4 pi sp1))
(setq ps15 (polar ps (/ pi 2) (+ co 0.20)))
(setq ps16 (polar ps15 0 0.15))
(command "circle" (polar ps (/ pi -2) (+ ra 0.0028)) (* 0.009 1.50))
(command "pline" ps16 ps15 (polar (polar ps (/ pi -2) (+ ra 0.0028)) (/ pi
2) (* 0.009 1.5))"")
(setq dial (* dial 1000))
(setq dial (rtos dial 2 0))
(setq spl (rtos spl 2 2))
(setq txl (strcat (strcat ch2 dial "mm @") spl""))
(command "text" (polar ps16 0 0.025) 0.035 0.0 txl)

(setq pa (polar pu4 pi (/ sp1 2)))
(setq ps17 (polar pa (/ pi 2) (+ co 0.10)))
(setq ps18 (polar ps17 0 0.15))
(command "pline" ps18 ps17 pa  (polar pa (/ pi 1.5) 0.025) pa  (polar
pa (/ pi 3) 0.025)"")

(setq di1 (getreal "\nDiameter steel positive moment
short span <mm>:"))
(if (< di1 12.0)(setq ch3 "RB")(setq ch3 "DB"))
(setq spc (getreal "\nSpacing steel <0.15>:"))
(if (not spc)(setq spc 0.15))
(setq spc (rtos spc 2 2))
(setq di1 (rtos di1 2 0))
(setq txz (strcat (strcat ch3 di1 "mm @") spc""))
(command "text" (polar ps12 0 0.025) 0.035 0.0 txz)

(setq dias (getreal "\nDiameter steel negative moment
short span <mm>:"))
```

```
                    (if (< dias 12.0)(setq ch4 "RB")(setq ch4 "DB"))

                        (setq spcs (getreal "\nSpacing steel <0.15>:"))
                    (if (not spcs)
                        (setq spcs 0.15)
                    )

                (setq dias (rtos dias 2 0))
                (setq spcs (rtos spcs 2 2))
                (setq txcs (strcat (strcat ch4 dias "mm @") spcs""))
            (command "text" (polar ps16 0 0.025) 0.035 0.0 txcs)

             (command "zoom"  "p")
             (initget 2 " 1 2 3 ")
                (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3 :"))
                (setq sc (atoi sca))
          (if (> sc 2)
                 (command "text" pd12 0.04 0.00 "1:25")
              (if (< sc 2)
                 (command "text" pd12 0.04 0.00 "1:10")
                 (command "text" pd12 0.04 0.00 "1:20")
              )
           )
                (cond
                   (( = sca "1") (command "scale" "w" pt9 pt10 "" bm 10))
                   (( = sca "2") (command "scale" "w" pt9 pt10 "" bm 5))
                   (( = sca "3") (command "scale" "w" pt9 pt10 "" bm 4))
                )
                   (princ)
) ;end
                    (setvar "blipmode" 1)
                    (setvar "cmdecho" 1)
```

BEAMS.LSP

```
                    (vmon)
(defun c:beams()
                    (setvar "BLIPMODE" 0)
                    (setvar "CMDECHO" 0)
                       (setq nam (getstring "\nName of beam:"))
                       (setq stp (getpoint "\nStart point at low left corner:"))
                       (setq wb (getdist stp "\nWidth of beam<0.20>:"))
                   (if (not wb)
                       (setq wb 0.20)
                   )
                       (setq hb (getdist stp "\nHeight of beam<0.40>:"))
                   (if (not hb)
                       (setq hb 0.40)
                   )

                       (setq co (getreal "\nConcrete covering <0.025>:"))
                         (if (not co) (setq co 0.025))
                     (while (or (< co 0.025)(> co 0.05))
                       (setq co (getreal "\nConcrete covering <0.025>:"))
                         (if (not co) (setq co 0.025))
                     )

                       (setq spc (getreal "\nSpacing steel of row<0.025>:"))
                         (if (not spc) (setq spc 0.025))
                    (while (or (< spc 0.025)(> spc 0.05))
                       (setq spc (getreal "\nSpacing steel of row<0.025>:"))
                         (if (not spc) (setq spc 0.025))
                    )

                       (setq pt1 (polar stp (/ pi 2) hb))
                       (setq pt2 (polar pt1 0 wb))
```

```
(setq pt3 (polar pt2 (+ pi (/ pi 2)) hb))
(setq pt4 (polar stp pi 0.07))
(setq pt5 (polar pt1 pi 0.07))
(setq pt6 (polar pt1 (/ pi 2) 0.07))
(setq pt7 (polar pt2 (/ pi 2) 0.07))
(setq pt8 (polar pt3 (/ pi 2) (/ hb 2)))
(setq pt9 (polar pt8 pi co))
(setq pt10 (polar pt8 0 (* wb 0.55)))
(setq pt11 (polar pt10 0 0.025))
(command "pline" stp "w" 0.00 stp"")
(command "pline" stp pt1 pt2 pt3 "c")
(command "offset" co stp (polar stp (/ pi 4) (* 1.414 co)) "")
(command "_pedit" (polar stp (/ pi 4) (* 1.414 co))  "w"
0.005"")
(command "pline" pt4 pt5 ""  "pline" pt6 pt7 "")
(command "pline" (polar pt4 0 -0.025) (polar pt4 0 0.025) "")
(command "pline" (polar pt5 0 -0.025) (polar pt5 0 0.025) "")
(command "pline" (polar pt6 (/ pi 2) -0.025) (polar pt6 (/ pi 2)
0.025) "")
(command "pline" (polar pt7 (/ pi 2) -0.025) (polar pt7 (/ pi 2)
0.025) "")
(command "pline" pt9 pt10 "")
(command "pline" (polar pt9 (/ pi 4) 0.02)  pt9 (polar pt9 (/ pi -
4) 0.02) "")
(setq pyy (polar pt4 (/ pi 2) (/ hb 2)))
(setq py (polar pyy pi 0.12))
(setq hb (rtos hb 2 2))
(command "text" py 0.03 0.00 hb )
(setq pxx (polar pt6 0 (/ wb 3.5)))
(setq px (polar pxx (/ pi 2) 0.03))
(setq wb (rtos wb 2 2))
(command "text" px 0.03 0.00 wb )
(setq wb (atof wb))
```

```
        (setq pt12 (polar pt1 (/ pi 1.3) (* wb 2.5)))
        (setq pt13 (polar pt3 (/ pi -6) (* wb 5)))
        (setq pd1 (polar stp 0 (/ wb 2)))
        (setq pd2 (polar pd1 (/ pi -2) 0.07))
        (setq pd3 (polar pd2 pi 0.17 ))
        (setq pd4 (polar pd3 (/ pi -2) 0.08))
        (setq pd5 (polar pd4 0 0.20))
        (setq pd6 (polar pd5 0 0.14))
        (setq pd7 (polar pd6 (/ pi 2) 0.08))
        (setq pd8 (polar pd7 pi 0.14))
    (command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5"")
        (setq pd9 (polar pd4 0 0.06))
        (setq pd10 (polar pd9 (/ pi 2) 0.015))
    (command "text" pd10 0.04 0.00 nam)
    (command "zoom"  "w" pt12 pt13 )
    (command "_pedit" pd2  "w" 0.005"")
        (setq ns1 (getint "\nNumber of stirrup steel<1> :"))
(if (not ns1)
    (setq ns1 1 )
)
        (setq ds1 (getreal "\nDiameter stirrup steel<6>:"))
(if (not ds1)(setq ds1 6 ))
(if (< ds1 12)(setq ch1 "RB")(setq ch1 "DB"))

        (setq sp1 (getreal "\nSpacing stirrup steel<0.20>:"))
(if (not sp1)
    (setq sp1 0.20)
)
        (setq css (itoa ns1))
        (setq dss (* ds1 1))
        (setq sp (rtos sp1 2 2))
        (setq ds (rtos dss 2 0))
```

```
                    (setq txo (strcat (strcat (substr css 1 1) ch1 ds "mm @")
sp"" ))
                (command "text" pt11 0.025 0.0 txo )
    (defun steel (pt1 ang1 ang2 tx)
                    (setq rs (/ (/ dis 2) 1000))
                    (setq dis (* rs 2))
                    (setq rco (+ co (+ rs 0.005)))
                    (setq ps1 (polar pt1 0.0 (- wb rco)))
                    (setq ps2 (polar ps1 (/ pi ang1) (+ rco y)))
                    (setq dxc (/ (- wb (* 2 rco)) (- nus 1)))
      (if (> (- dxc (* rs 2)) 0.025)
       (progn
                    (setq x 0)
          (repeat nus
                    (setq ps3 (polar ps2 pi x))
                    (setq x (+ x dxc))
                (command "donut" "0.0" dis ps3 "")
                    (setq ps4 (polar ps3 (/ pi ang2) 0.035))
                    (setq ps5 (polar ps4 0 (* wb 0.65)))
                (command "pline" ps3 ps4 ps5 "")
          )
                    (setq cn (itoa nus))
                    (setq dis (* dis 1000))
                 (if (< dis 12)(setq ch2 "RB")(setq ch2 "DB"))
                    (setq cs (rtos dis 2 0))
                    (setq txt (strcat (substr cn 1 1) ch2 cs "mm"))
                    (setq ptx (polar ps5 0 (* wb 0.80)))
                (command "text" ptx tx 0.0 txt)
                    (setq y (+ y (+ spc  (* 2 rs))))
       )
      )
    );steel
                (setq y 0.0)
```

```
                (setq nus 1)
            (while nus
                (progn
                        (setq nus (getint "\nNumber of lower steel :"))
                    (if nus
                        (progn
                            (setq dis (getreal "\nDiameter steel< mm >:"))
                            (steel stp 2 3 0.025)
                        )
                    )
                )
            )
                (setq y 0.0)
                (setq nus 1)
            (while nus
                (progn
                        (setq nus (getint "\nNumber of upper steel :"))
                    (if nus
                        (progn
                            (setq dis (getreal "\nDiameter steel< mm >:"))
                            (steel pt1 -2 -3 0.025)
                        )
                    )
                )
            )
                (command "zoom"  "p" )
                        (setq pd11 (polar pd5 0 0.02))
                        (setq pd12 (polar pd11 (/ pi 2) 0.015))
            (initget 2 " 1 2 3 ")
                        (setq sca (getkword "\nScale1:10=1,1:20=2,1:25=3
:"))
                        (setq sc (atoi sca))
                    (if (> sc 2)
```

```
                                (command "text" pd12 0.04 0.00 "1:25")
                            (if (< sc 2)
1.                              (command "text" pd12 0.04 0.00 "1:10")
                                (command "text" pd12 0.04 0.00 "1:20")
                            )
                        )
                        (cond
                            (( = sca "1") (command "scale"  "w" pt12 pt13 "" stp 10))
                            (( = sca "2") (command "scale"  "w" pt12 pt13 "" stp 5))
                            (( = sca "3") (command "scale"  "w" pt12 pt13 "" stp 4))
                        )
                (princ)
                (princ)
        );beams
                (SETVAR "BLIPMODE" 1)
                (SETVAR "CMDECHO" 1)
```

BEAMSIDE.LSP

```
(vmon)
(defun c:bs ()
                (setvar "blipmode" 0)
                (setvar "cmdecho" 0)
                        (setq nam (getstring "\nName of beam :"))
                        (setq bm (getpoint "\nStart point left support of beam
:"))
                        (setq len (getdist bm "\nLength of beam<3.00>:"))
                    (if (not len) (setq len 3.00))
                (setq tbm (getreal "\nDepth of beam  <0.50>:"))
                    (if (not tbm)(setq tbm 0.50))
                        (setq lenl (getdist bm "\nLeft Cantilever<1.00>:"))
                    (if (not lenl) (setq lenl 1.00))
                (setq tbl (getreal "\nDepth of beam <0.30>:"))
                    (if (not tbl) (setq tbl 0.30))
                        (setq lenr (getdist bm "\nRiht Cantilever<1.20>:"))
                    (if (not lenr) (setq lenr 1.20))
                (setq tbr (getreal "\nDepth of beam <0.40>:"))
                    (if (not tbr) (setq tbr 0.40))
                 (setq co (getreal "\nConcrete Covering <0.025>:"))
                    (if (not co)(setq co 0.025))
                        (setq wc (getreal "\nWidth of column <0.20>:"))
                  (if (not wc)(setq wc 0.20))

            (if (= lenl 0.00)(setq lenl (/ wc 2)))
            (if (= lenr 0.00)(setq lenr (/ wc 2)))

                (setq bm1 (polar bm 0 len))
(setq bm1 (polar bm 0 len))
                (setq pt1 (polar bm 0 (/ wc 2)))
                (setq pt2 (polar pt1 0 (- len wc)))
```

```
(setq pt3 (polar bm1 0 (/ wc 2)))
(setq pt4 (polar pt3 (/ pi -2) 0.30))
(setq pt5 (polar pt3 0 (- lenr (/ wc 2))))
(setq pt6 (polar pt5 (/ pi 2) tbr))
(setq pt7 (polar pt6 pi (- lenr (/ wc 2))))
(setq pt8 (polar pt7 (/ pi 2) (+ (- tbm tbr) 0.30)))
(setq pt9 (polar pt2 (/ pi 2) tbm))
(setq pt10 (polar pt1 (/ pi 2) tbm))
(setq pt11 (polar pt1 pi wc))
(setq pt12 (polar pt11 pi (- lenl (/ wc 2))))
(setq pt13 (polar pt12 (/ pi 2) tbl))
(setq pt14 (polar pt13 0 (- lenl (/ wc 2))))
(setq pt15 (polar pt14 (/ pi 2) (+ (- tbm tbl) 0.30)))
(setq pt16 (polar pt11 (/ pi -2) 0.30))
(setq pt17 (polar pt15 (- pi (/ pi 6)) (+ (* lenl 2) 0.80)))
(setq pt18 (polar pt4 (/ pi -4) (+ (* lenr 2) 0.80)))
(command "pline" pt15 pt14 pt13 pt12 pt11 pt16""
         "pline" (polar pt1 (/ pi -2) 0.30) pt1 pt2 (polar pt2 (/ pi -2)
0.30)""
         "pline" pt4 pt3 pt5 pt6 pt7 pt8""
         "pline" (polar pt9 (/ pi 2) 0.30) pt9 pt10 (polar pt10 (/ pi 2)
0.30)"") ; line concrete

(setq l2 (/ len 4))
(setq px1 (polar bm (/ pi -2) 0.50))
(setq px2 (polar bm1 (/ pi -2) 0.50))
(setq px3 (polar px1 0 (/ len 2)))
(setq px4 (polar px3 (/ pi -2) 0.07))
(setq px5 (polar px1 (/ pi 2) 0.05))
(setq px6 (polar px5 0 l2))
(setq px7 (polar px5 0 (/ l2 2.5)))
(setq px8 (polar px7 (/ pi 2) 0.025))
(setq px9 (polar px8 0 (- len l2)))
```

```
(command "pline" (polar px1 pi (+ lenl 0.025)) px1 px2 (polar px2 0 (+ lenr
0.025))"" ; line mid span
        "pline" (polar px1 (/ pi 2) 0.075) (polar px1 (/ pi -2) 0.025)""
        "pline" (polar px2 (/ pi 2) 0.075) (polar px2 (/ pi -2) 0.025)"")

        (command "pline" (polar px5 pi 0.025) (polar px6 0 0.025)""
                "pline" (polar px6 (/ pi 2) 0.025) (polar px6 (/ pi -2)
0.025)""
                "text" px4 0.03 0.00 (rtos len 2 2) ; mid span
                "text" px8 0.03 0.00 (rtos l2 2 2)   ; letf span
                "text"  px9 0.03 0.00 (rtos l2 2 2)) ; right span


          (if (/= lenl (/ wc 2))
                (command "pline" (polar (polar px1 pi lenl) (/ pi 2) 0.025)
(polar (polar px1 pi lenl) (/ pi -2) 0.025)""
                          "text" (polar px4 pi (+ (/ len 2) (/ lenl 2))) 0.03
0.00 (rtos lenl 2 2)))
          (if (/= lenr (/ wc 2))
                (command "pline" (polar (polar px2 0 lenr) (/ pi 2) 0.025)
(polar (polar px2 0 lenr) (/ pi -2) 0.025)""
                          "text" (polar px4 0 (+ (/ len 2) (/ lenr 2))) 0.03
0.00 (rtos lenr 2 2)))


        (if (= lenr (/ wc 2))
            (progn
                (setq py1 (polar pt9 0 (+ wc 0.10)))
                (setq py2 (polar py1 (/ pi -2) tbm))
                (setq py3 (polar py1 (/ pi -2) (/ tbm 2)))
                (setq py4 (polar py3 0 0.05))
                (command  "text" py4 0.03 0.00 (rtos tbm 2 2))
```

```
                    (progn
                        (setq py1 (polar pt6 0 0.10))
                        (setq py2 (polar py1 (/ pi -2) tbr))
                        (setq py3 (polar py1 (/ pi -2) (/ tbr 2)))
                        (setq py4 (polar py3 0 0.05))
                        (command  "text" py4 0.03 0.00 (rtos tbr 2 2))
                    )
            ) ;if
                                (command "pline" (polar py1 (/ pi 2) 0.025) py1
py2 (polar py2 (/ pi -2) 0.025)""
                                                    "pline" (polar py1 pi 0.03) (polar py1
0 0.025)""
                                                    "pline" (polar py2 pi 0.03) (polar py2
0 0.025)"")

    (if (/= lenl (/ wc 2))
        (progn
            (command "pline" (polar (polar pt13 pi 0.10) (/ pi 2) 0.025) (polar
(polar pt13 pi 0.10) (/ pi -2) (+ tbl 0.025))""
                        "pline" (polar (polar pt13 pi 0.10) pi 0.025)  (polar (polar
pt13 pi 0.10) 0 0.025)""
                        "pline" (polar pt12 pi 0.075) (polar pt12 pi 0.125)""
                        "text"  (polar  (polar (polar pt13 pi 0.10) (/ pi -2) (/ tbl 2))
pi 0.12)  0.03 0.00 (rtos tbl 2 2))
        )
    ) ;if

                (setq pd1 px4)
                (setq pd2 (polar pd1 (/ pi -2) 0.05))
                (setq pd3 (polar pd2 pi 0.17))
                (setq pd4 (polar pd3 (/ pi -2) 0.08))
                (setq pd5 (polar pd4 0 0.20))
                (setq pd6 (pclar pd5 0 0.14))
```

```
                (setq pd7 (polar pd6 (/ pi 2) 0.08))
                (setq pd8 (polar pd7 pi 0.14))
                (setq pd9 (polar pd4 0 0.06))
                (setq pd10 (polar pd9 (/ pi 2) 0.015))
                (setq pd11 (polar pd5 0 0.02))
                (setq pd12 (polar pd11 (/ pi 2) 0.015))
            (command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5 "")
            (command "text" pd10 0.04 0.00 nam)

            (command "zoom" "w" pt17 pt18 )

            (command "mirror"  px7 (polar px6 (/ pi 2) 0.025) "" px3 (polar px3 (/
pi 2) 0.50) "")
            (command "_pedit" pd2 "w" 0.005 "")

                    (setq psl1 (polar bm (/ pi 2) co))
                    (setq psl2 (polar bm (/ pi 2)(- tbl co)))
                    (setq psr1 (polar bm1 (/ pi 2) co))
                    (setq psr2 (polar bm1 (/ pi 2)(- tbr co)))
                    (setq ps1 (polar psl1 pi (- lenl (+ co 0.015))))
                (setq ps2 (polar psr1 0 (- lenr (+ co 0.015))))
                    (setq ps3 (polar psl2 pi (- lenl (+ co 0.015))))
                    (setq ps4 (polar psr2 0 (- lenr (+ co 0.015))))
                (cond
                    ((= tbl tbm) (setq psl3 (polar ps3 0 (- (/ wc 2) (+ co
0.015)))))
                    ((< tbl tbm) (setq psl3 (polar bm (/ pi 2)(- tbm co))))
                    ((> tbl tbm) (setq psl3 (polar (polar ps3 0 (- (/ wc 2) (+ co
0.015))) (/ pi -2) (- tbl tbm))))
                )
                    (cond
                    ((= tbr tbm) (setq psr3 (polar ps4 pi (- (/ wc 2) (+ co
0.015)))))
```

```
                    ((< tbr tbm) (setq psr3 (polar bm1 (/ pi 2)(- tbm co))))
                    ((> tbr tbm) (setq psr3 (polar (polar ps4 pi (- (/ wc 2) (+ co
0.015))) (/ pi -2) (- tbr tbm))))
                )

                    (command "pline" (polar (polar psl3 pi (- (/ wc 2) (+ co
0.015))) (/ pi -4)(* 1.414 0.03))
                        (polar (polar psl3 pi (- (/ wc 2) (+ co 0.015))) (/ pi -2)
0.03) "a"
                        (polar psl3 pi (- (/ wc 2) (+ co 0.015))) "l" (polar psr3 0
(- (/ wc 2) (+ co 0.015))) "a"
                        (polar (polar psr3 0 (- (/ wc 2) (+ co 0.015))) (/ pi -2)
0.03) "l"
                        (polar (polar psr3 0 (- (/ wc 2) (+ co 0.015))) (- (/ pi 4)
pi)(* 1.414 0.03))"") ;upper mid steel
        (if (/= lenl (/ wc 2))
            (if (< tbl tbm)
              (command "pline" (polar ps3 (/ pi -4)(* 1.414  0.03)) (polar ps3 (/
pi -2) 0.03) "a" ps3 "l"
                (polar ps3 0 (- (+ lenl l2) (+ co 0.015))) "a" (polar (polar ps3 0 (-
(+ lenl l2) (+ co 0.015))) (/  pi  -2) 0.03) "l"
                (polar (polar (polar ps3 0 (- (+ lenl l2) (+ co 0.015))) (/ pi -2) 0.03)
pi 0.03)"") ;upper left steel
            )
        )
          (if (/= lenl (/ wc 2))
            (if (> tbl tbm)
              (command "pline" (polar ps3 (/ pi -4)(* 1.414 0.03)) (polar ps3 (/
pi -2) 0.03) "a" ps3 "l"
                (polar ps3 0 (- (+ lenl (/ wc 2)) (* co 3))) "a" (polar (polar ps3 0 (-
(+ lenl (/ wc 2)) (* co 3))) (/ pi -2) 0.03) "l"
                (polar (polar (polar ps3 0 (- (+ lenl (/ wc 2)) (* co 3))) (/ pi -2)
0.03) pi 0.03)"") ;upper left new steel
```

```
                )
              )

                (if (/= lenr (/ wc 2))

                    (if (< tbr tbm)
                        (command "pline"  (polar ps4 (- (/ pi 4) pi)(* 1.414 0.03))
(polar ps4 (/ pi -2) 0.03)  "a" ps4 "l"
                        (polar ps4 pi (- (+ lenr l2) (+ co 0.015))) "a"  (polar (polar
ps4 pi (- (+ lenr l2) (+ co 0.015)))(/ pi -2) 0.03)
                        "l"  (polar (polar ps4 pi (- (+ lenr l2) (+ co 0.015))) (/ pi -4)(*
1.414 0.03))"") ;upper right steel
                    )
                )
            (if (/= lenr (/ wc 2))
                (if (> tbr tbm)
                    (command "pline"  (polar ps4 (- (/ pi 4) pi)(* 1.414 0.03)) (polar
ps4 (/ pi -2) 0.03)  "a" ps4 "l"
                    (polar ps4 pi (- (+ lenr (/ wc 2)) (* co 3)))  "a"  (polar (polar ps4
pi (- (+ lenr (/ wc 2)) (* co 3))) (/ pi -2) 0.03)
                    "l"  (polar (polar ps4 pi (- (+ lenr (/ wc 2)) (* co 3))) (/ pi -4)(*
1.414 0.03))"") ;upper right new steel
                )
            )

    (command "pline" (polar ps1 (/ pi 4)(* 1.414 0.03))  (polar ps1 (/ pi 2) 0.03)
"a" ps1 "l" ps2 "a"
                (polar ps2 (/ pi 2) 0.03) "l"  (polar ps2 (- pi (/ pi 4))(* 1.414
0.03))"") ;lower mid steel

        (setq s 1)
        (setq so 3)
      (while (<= s so)
```

```
(initget 2 " L M R  ")
(setq stir (getkword "\nStirrup steel( L, M, R ) <no> :"))
(cond

((= stir "L")    (setq ds1 (getreal "\nDiameter steel stirrup
Left cantilever <6> :"))
(setq sp1 (getreal "\nSpacing steel <0.20> :"))
(setq p1 (polar psl1 pi (- lenl (+ co 0.03))))
(setq p2 (polar psl1 pi (+ (/ wc 2) 0.05)))

(setq p3 (polar p2 (/ pi 2)(- tbl (* co 2))))

(if (not ds1) (setq ds1 6.0))

(if (not sp1) (setq sp1 0.20))
(setq hi (distance p2 p3))
(setq x1 (- (car p2)(car p1)))
(setq ro (/ x1 sp1))
(if (/= ro (fix ro))
(setq ro (1+ (fix ro))))
(setq sp1 (/ x1 ro))
(setq i 1)
(while (<= i ro)
(setq p2 (polar p1 (/ pi 2) hi))
(command "pline" p1 p2 "")
(setq p1 (polar p1 0 sp1))
(setq i (1+ i)))
(command "pline" p1 (polar p1 (/ pi
2) hi)""))

((= stir "M")    (setq ds1 (getreal "\nDiameter steel stirrup Middle
span <6> :"))
(setq sp1 (getreal "\nSpacing steel <0.20> :"))
(setq p1 (polar psl1 0 (+ (/ wc 2) 0.10)))
```

```
                    (setq p2 (polar psr1 pi (+ (/ wc 2) 0.10)))
                    (setq p3 (polar p2 (/ pi 2)(- tbm (* co 2))))

                    (if (not ds1) (setq ds1 6.0))

                                (if (not sp1) (setq sp1 0.20))
                                (setq hi (distance p2 p3))
                            (setq x1 (- (car p2)(car p1)))
                            (setq ro (/ x1 sp1))
                               (if (/= ro (fix ro))
                                  (setq ro (1+ (fix ro))))
                                  (setq sp1 (/ x1 ro))
                                  (setq i 1)
                            (while (<= i ro)
                                  (setq p2 (polar p1 (/ pi 2) hi))
                                  (command "pline" p1 p2 "")
                                  (setq p1 (polar p1 0 sp1))
                                  (setq i (1+ i)))
                                  (command "pline" p1 (polar p1 (/ pi
2) hi)""))
                ((= stir "R")   (setq ds1 (getreal "\nDiameter steel stirrup
Right cantilever <6> :"))
                    (setq sp1 (getreal "\nSpacing steel <0.20> :"))
                        (setq p1 (polar psr1 0 (+ (/ wc 2) 0.05)))
                    (setq p2 (polar psr1 0 (- lenr (+ co 0.03))))
                    (setq p3 (polar p2 (/ pi 2)(- tbr (* co 2))))
                    (if (not ds1) (setq ds1 6.0))

                                (if (not sp1) (setq sp1 0.20))
                                (setq hi (distance p2 p3))
                            (setq x1 (- (car p2)(car p1)))
                            (setq ro (/ x1 sp1))
                               (if (/= ro (fix ro))
```

```
                                        (setq ro (1+ (fix ro))))
                                        (setq sp1 (/ x1 ro))
1.                                      (setq i 1)
                                    (while (<= i ro)
                                        (setq p2 (polar p1 (/ pi 2) hi))
                                        (command "pline" p1 p2 "")
                                        (setq p1 (polar p1 0 sp1))
                                        (setq i (1+ i)))
                                        (command "pline" p1 (polar p1 (/ pi
2) hi)""))
                                ((null stir))
                            ) ;cond

                        (setq s (1+ s))
                    ) ;while

                    (command "zoom"  "p")
                    (initget 2 " 1 2 3  ")
                        (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3  :"))
                        (setq sc (atoi sca))
                    (if (> sc 2)
                            (command "text" pd12 0.04 0.00 "1:25")
                        (if (< sc 2)
                            (command "text" pd12 0.04 0.00 "1:10")
                            (command "text" pd12 0.04 0.00 "1:20")
                        )
                    )
                        (cond
                            (( = sca "1") (command "scale" "w" pt17 pt18 "" bm 10))
                            (( = sca "2") (command "scale" "w" pt17 pt18 "" bm 5))
                            (( = sca "3") (command "scale" "w" pt17 pt18 "" bm 4))
                        )
                            (princ\
```

```
) ;end bs
                    (setvar "blipmode" 1)
                    (setvar "cmdecho" 1)
```

CANBEAM.LSP

```
(vmon)
 (defun c:cb()
                    (setvar "BLIPMODE" 0)
                    (setvar "CMDECHO" 0)
                    (setq nam (getstring "\nName of Cantilever Beam :"))
                    (setq bm (getpoint "\nBase point under beam:"))
                    (setq xd (getdist bm "\nLength of cantilever beam
<1.00>:"))
                  (if (not xd)(setq xd 1.00))
                    (setq yd (getdist bm "\nHigh of beam <0.40>:"))
                (if (not yd)(setq yd 0.40))
                  (setq y1 (getreal  "\nLength of steel at support  <0.70>:"))
                (if (not y1)(setq y1 0.70))
                    (setq wc (getreal  "\nColumn width <0.20>:"))
                (if (not wc)(setq wc 0.20))
                  (setq co (getreal  "\nCovering of concrete <0.025>:"))
                (if (not co)(setq co 0.025))

                    (setq pt1 (polar bm 0 (/ wc 2)))
                    (setq pt2 (polar bm 0 xd))
                    (setq pt3 (polar pt2 (/ pi 2) yd))
                  (setq pt4 (polar pt1 (/ pi 2) yd))
                    (setq pt5 (polar pt4 (/ pi 2) 0.30))
               (setq pt6 (polar pt5 pi wc))
               (setq pt7 (polar pt6 (/ pi -2)(+ (* 0.30 2) y1)))
               (setq pt8 (polar pt7 0 wc))
                    (setq pt9 (polar pt6 (- pi (/ pi 4)) 0.50))
               (setq pt10 (polar pt8 (/ pi -6) (+ xd 0.50)))
              (command "pline" pt8 pt1 pt2 pt3 pt4 pt5 "")
              (command "pline" pt6 pt7 "")
```

```
(command "pline" (polar pt6 pi 0.025) (polar pt6 0 (/ (- wc 0.07) 2)) (polar
(polar pt6 0 (/ (- wc 0.07) 2)) (/ pi 2) 0.025)
                    (polar (polar pt5 pi (/ (- wc 0.07) 2)) (/ pi -2) 0.025)
(polar pt5 pi (/ (- wc 0.07) 2)) (polar pt5 0 0.025)""
                "pline" (polar pt7 pi 0.025) (polar pt7 0 (/ (- wc 0.07) 2)) (polar
(polar pt7 0 (/ (- wc 0.07) 2)) (/ pi 2) 0.025)
                    (polar (polar pt8 pi (/ (- wc 0.07) 2)) (/ pi -2) 0.025)
(polar pt8 pi (/ (- wc 0.07) 2)) (polar pt8 0 0.025)"")


                    (setq px1 (polar (polar pt7 0 (/ wc 2))(/ pi -2) 0.10))
                    (setq px2 (polar px1 0 xd))
                    (setq px3 (polar px1 0 (/ xd 2)))
                    (setq px4 (polar px3 (/ pi -2) 0.07))
                (command "pline" (polar px1 pi 0.025) px1 px2 (polar px2 0
0.025)"")
                (command "pline" (polar px1 (/ pi 2) 0.025) (polar px1 (/ pi -
2) 0.025)"")
                (command "pline" (polar px2 (/ pi 2) 0.025) (polar px2 (/ pi -
2) 0.025)"")
                (command "text" px4 0.035 0.00 (rtos xd 2 2))


                    (setq py1 (polar pt3 0 0.10))
                    (setq py2 (polar py1 (/ pi -2) yd))
                    (setq py3 (polar py1 (/ pi -2) (/ yd 2)))
                    (setq py4 (polar py3 0 0.05))
                    (setq py5 (polar (polar pt4 pi (+ wc 0.10)) (/ pi -2) co))
                    (setq py6 (polar py5 (/ pi -2) y1))
                    (setq py7 (polar py5 (/ pi -2) (/ y1 2)))
                    (setq py8 (polar py7 pi 0.15))
                (command "pline" (polar py1 (/ pi 2) 0.025) py1 py2 (polar
py2 (/ pi -2) 0.025)"")
```

```
(command "pline" (polar py1 pi 0.03) (polar py1 0 0.025)"")
(command "pline" (polar py2 pi 0.03) (polar py2 0 0.025)"")
(command "text" py4 0.035 0.00 (rtos yd 2 2))
(command "pline" (polar py5 (/ pi 2) 0.025) py5 py6 (polar
py6 (/ pi -2) 0.025)"")
(command "pline" (polar py5 pi 0.03) (polar py5 0 0.025)"")
(command "pline" (polar py6 pi 0.03) (polar py6 0 0.025)"")
(command "text" py8 0.035 0.00 (rtos y1 2 2))


(setq pd1 px4)
(setq pd2 (polar pd1 (/ pi -2) 0.05))
(setq pd3 (polar pd2 pi 0.17))
(setq pd4 (polar pd3 (/ pi -2) 0.08))
(setq pd5 (polar pd4 0 0.20))
(setq pd6 (polar pd5 0 0.14))
(setq pd7 (polar pd6 (/ pi 2) 0.08))
(setq pd8 (polar pd7 pi 0.14))
(setq pd9 (polar pd4 0 0.06))
(setq pd10 (polar pd9 (/ pi 2) 0.015))
(setq pd11 (polar pd5 0 0.02))
(setq pd12 (polar pd11 (/ pi 2) 0.015))
(command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5 "")
(command "text" pd10 0.04 0.00 nam)


(command "zoom" "w" pt9 pt10 )
(command "_pedit" pd2 "w" 0.005 "")


(setq ps1 (polar bm (/ pi 2) co))
(setq ps2 (polar ps1 pi (- (/ wc 2) (* co 3))))
(setq ps3 (polar ps1 0 (- xd (+ co 0.015))))
(setq ps4 (polar ps3 (/ pi 2) (- yd (* co 2))))
(setq ps5 (polar ps4 pi (+ xd (/ wc 2))(+ (* co 2) 0.015))))
```

```
(setq ps6 (polar ps5 (/ pi -2) y1))
(setq ps7 (polar ps1 0 (+ (/ wc 2) 0.05)))
(command "pline" (polar ps6 (/ pi 4)(* 1.414 0.03)) (polar ps6 0
0.03) "a" ps6 "l" ps5 ps4 "a"
(polar ps4 (/ pi -2) 0.03) (polar ps4 (- (/ pi 4) pi)(*
1.414 0.03))"")
(command "pline" (polar (polar ps5 0 (- (+ wc 0.10) co)) (- pi (/ pi 3))
0.025) (polar ps5 0 (- (+ wc 0.10) co))
(polar (polar ps5 0 (- (+ wc 0.10) co)) (/ pi 2) (+ co
0.15))
(polar (polar (polar ps5 0 (- (+ wc 0.10) co)) (/ pi 2) (+
co 0.15)) 0 0.15)"")

(setq diat (getreal "\nDiameter Tension steel <16>:"))
(if (not diat) (setq diat 16.00))
(setq ns1 (getreal "\nNumber steel <2>:"))
(if (not ns1) (setq ns1 2))

(setq txt (strcat (substr (rtos ns1 2 0) 1 2)
" %%c" (rtos diat 2 0) "mm"))
(command "text" (polar (polar (polar ps5 0 (- (+ wc 0.10) co)) (/ pi 2) (+
co 0.15)) 0 0.17) 0.035 0.0 txt)

(command "pline" (polar ps2 (/ pi 4)(* 1.414 0.03)) (polar ps2
(/ pi 2) 0.03) "a" ps2 "l" ps3 "a"
(polar ps3 (/ pi 2) 0.03) "l" (polar ps3 (- pi (/ pi 4))(*
1.414 0.03))"")
(command "pline" (polar (polar ps1 0 (+ (/ wc 2) 0.10)) (- (/ pi 3) pi)
0.025) (polar ps1 0 (+ (/ wc 2) 0.10))
(polar (polar ps1 0 (+ (/ wc 2) 0.10)) (/ pi -2) (+ co
0.15))
```

```
                    (polar (polar (polar ps1 0 (+ (/ wc 2) 0.10)) (/ pi -2) (+
co 0.15)) 0 0.15)"")

                                (setq dial (getreal "\nDiameter
Compression steel <12>:"))
                                (if (not dial) (setq dial 12.00))
                                (setq ns2 (getreal "\nNumber steel <2>:"))
                                (if (not ns2) (setq ns2 2))

                                (setq txc (strcat (substr (rtos ns2 2 0) 1 2)
" %%c" (rtos dial 2 0) "mm"))
        (command "text" (polar (polar (polar ps1 0 (+ (/ wc 2) 0.10)) (/ pi -2) (+
co 0.15)) 0 0.17) 0.035 0.0 txc)

                    (setq dia (getreal "\nDiameter Stirrup steel <6>:"))
                (if (not dia) (setq dia 6.00))
                    (setq sp (getreal "\nSpacing steel <0.20>:"))
                (if (not sp)  (setq sp 0.20))
                    (setq dia (/ dia 1000))
                    (setq ra (/ dia 2))
                    (setq x1 (- xd (+ (+ (/ wc 2) 0.05)(+ co 0.015))))
                    (setq hi (- yd (* co 2)))
                    (setq ns (/ x1 sp))
            (if (= ns (fix ns))
                    (setq ro ns)
                    (setq ro (+ (fix ns) 1))
            )
                    (setq sp (/ x1 ro))
                    (setq i 1)
                (while (<= i ro)
                    (command "pline" ps7 (polar ps7 (/ pi 2) hi)"")
                    (setq ps7 (polar ps7 0 sp))
```

```
                (setq i (+ i 1))
              )
                (command "pline" ps7 (polar ps7 (/ pi 2) hi)"")

                (command "zoom"  "p")

    (initget 2 " 1 2 3 ")
        (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3 :"))
        (setq sc (atoi sca))
  (if (> sc 2)
          (command "text" pd12 0.04 0.00 "1:25")
      (if (< sc 2)
          (command "text" pd12 0.04 0.00 "1:10")
          (command "text" pd12 0.04 0.00 "1:20")
      )
   )
      (cond
          (( = sca "1") (command "scale" "w" pt9 pt10 "" bm 10))
          (( = sca "2") (command "scale" "w" pt9 pt10 "" bm 5))
          (( = sca "3") (command "scale" "w" pt9 pt10 "" bm 4))
      )
  (princ)
) ;end
```

STAIRS.LSP

```
(vmon)
 (defun c:stairs ()
                    (setvar "blipmode" 0)
                    (setvar "cmdecho" 0)
                    (setq nam (getstring "\nName of stair:"))
                    (setq bm (getpoint "\nSelect base point:"))
                    (setq hi (getreal "\nFloor to Floor Height <1.50>:"))
              (if (not hi)(setq hi 1.50))
                  (setq ris (getreal "\nAssume Riser <0.175>:"))
              (if (not ris)(setq ris 0.175))
                  (setq tr (getreal "\nTread <0.25>:"))
              (if (not tr)(setq tr 0.25))
                  (setq t (getreal "\nThickness stair<0.12>:"))
              (if (not t)(setq t 0.12))
                  (setq ld (getreal "\nWidth Landing low <1.20> :"))
              (if (not ld)(setq ld 1.20))
                  (setq up (getreal "\nWidth Landing up <1.20>:"))
              (if (not up)(setq up 1.20))

                    (setq ze1 (atan ris tr))
                    (setq ze2 (/ ze1 2))
                    (setq zd (/ t (cos ze2)))
                    (setq pt3 (polar bm pi (+ ld 0.10)))
                    (setq pt4 (polar pt3 (/ pi -2) 0.40))
                    (setq pt5 (polar pt4 0 0.20))
                    (setq pt6 (polar pt5 (/ pi 2) (- 0.40 t)))
                    (setq pt7 (polar bm (+ (+ pi (/ pi 2)) ze2) zd))
                (command "pline" bm "w" 0.000 "" pt3 pt4 pt5 pt6 pt7 "")
                    (setq num (/ hi ris))
            (if (/= num (fix num))
                    (setq num (fix num))
```

```
)
        (setq ris (/ hi num))

        (setq pv1 (polar bm pi 0.15))
        (setq pv2 (polar pv1 (/ pi 2) ris))
        (setq pv3 (polar pv2 (/ pi 2) ris))
        (setq pv4 (polar bm (/ pi 2)(* ris 5)))
        (setq pv5 (polar pv4 0 tr))
        (setq pv6 (polar pv5 0 tr))
        (setq pv7 (polar pv3 0 (+ 0.15 (* tr 1.5))))
        (setq pv8 (polar pv7 (/ pi -2) 0.025))
        (setq pv9 (polar pv8 pi (- (/ tr 2)(+ 0.025 0.009))))
    (command "pline" pv1 (polar pv3 (/ pi 2) 0.025) "")
    (command "pline" (polar pv3 pi 0.025)(polar pv3 0 0.025) "")
    (command "pline" (polar pv2 pi 0.025)(polar pv2 0 0.025) "")
    (command "pline" (polar pv4 pi 0.025)(polar pv6 0 0.025) "")
    (command "pline" (polar pv4 (/ pi 2) 0.025)(polar pv4 (/ pi -2) 0.025)
"")
    (command "pline" (polar pv5 (/ pi 2) 0.025)(polar pv5 (/ pi -2) 0.025)
"")
    (command "pline" (polar pv6 (/ pi 2) 0.025)(polar pv6 (/ pi -2) 0.025)
"")
    (command "text" (polar (polar pv4 0 (/ tr 2.5))(/ pi 2) 0.025) 0.035 0.00
(rtos tr 2 2))
    (command "text" (polar (polar pv5 0 (/ tr 2.5))(/ pi 2) 0.025) 0.035 0.00
(rtos tr 2 2))
    (command "text" (polar (polar pv1 (/ pi 2)(/ ris 2)) pi 0.15) 0.035 0.00
(rtos ris 2 3))
    (command "text" (polar (polar pv2 (/ pi 2)(/ ris 2)) pi 0.15) 0.035 0.00
(rtos ris 2 3))
    (command "pline" (polar pv8 (- pi (/ pi 3)) 0.025) pv8 (polar pv8 (/ pi
3) 0.025) "")
```

```
(command "pline" pv8 (polar pv8 (/ pi 2) 0.30)(polar (polar pv8 (/ pi 2)
0.30) pi (+ 0.15 (* tr 1.5))) "")
1.          (command "pline" (polar pv9 (- pi (/ pi 3)) 0.025) pv9 (polar pv9 (/ pi
3) 0.025) "")
            (command "pline" pv9 (polar pv9 (/ pi 2) 0.15)(polar (polar pv9 (/ pi 2)
0.15) pi (+ 0.15 tr 0.025 0.009)) "")


                (setq i 1)
            (while (<= i num)
                    (setq p1 (car bm))
                    (setq p2 (cadr bm))
                    (setq pt1 (list p1 (+ p2 ris)))
                    (setq pt2 (list (+ p1 tr) (+ p2 ris)))
                (command "pline" bm pt1 pt2 "")
                    (setq bm pt2)
                    (setq i (+ 1 i))
            )
                (command "pline" bm pt1 "")
                    (setq pt0 (polar pt1 0 tr))
                    (setq pt8 (polar pt1 0 (+ up 0.10)))
                    (setq pt9 (polar pt8 (/ pi -2) 0.40))
                    (setq pt10 (polar pt9 pi 0.20))
                    (setq pt11 (polar pt10 (/ pi 2) (- 0.40 t)))
                    (setq pt12 (polar pt0 (+ (+ pi (/ pi 2)) ze2) zd))
                (command "pline" pt1 pt8 pt9 pt10 pt11 pt12 pt7 "")
                    (setq pt13 (polar pt11 (atan 0.025 0.10) (/ 0.10 (cos (atan
0.025 0.10)))))
                    (setq pt14 (polar pt13 pi (- up 0.07)))
                    (setq pt15 (polar pt4 (+ pi (/ pi 4)) 1.0))
                    (setq pt16 (polar pt8 (/ pi 4) 1.0))
                (command "pline" pt13 pt14 "a" (polar pt14 (/ pi 2) 0.03) (polar
pt14 (/ pi 4) (* 1.414 0.03)) "")
```

```
                (command "pline" pt13 "a" (polar pt13 (/ pi 2) 0.03) (polar pt13
(- pi (/ pi 4)) (* 1.414 0.03)) "")
                    (setq pta (polar pt3 (/ pi -4) (* 1.414 0.025)))
                    (setq ptaa (polar pta 0 (+ ld 0.10)))
                    (setq ptaaa (polar pta (/ pi -2) 0.30))
                    (setq ptaaaa (polar ptaa (/ pi -2) (- t 0.025 0.025 0.035)))
                (command "pline" ptaa "w" 0.000 "" pta ptaaa "a" (polar ptaaa
0 0.03) (polar ptaaa (/ pi 4) (* 1.414 0.03)) "")
                (command "pline" ptaa "w" 0.00 "" ptaaaa "a" (polar ptaaaa 0
0.02) (polar ptaaaa (/ pi 4) (* 1.414 0.02)) "")
                    (setq ptz1 (polar ptaa 0 (/ (- t (* 1.414 0.025) 0.025 0.015)
(sin ze1))))
                    (setq ptz2 (polar ptz1 (atan ris tr) 0.30))
                (command "pline" ptaa ptz1 ptz2 "a" (polar ptz2 (+ (/ pi 2) ze1)
0.03) (polar ptz2 (+ (/ pi 2) ze1 (/ pi 4)) (* 1.414 0.03)) "")

            (setq i 1)
        (while (<= i num)
                    (setq ptt1 (polar ptaa (/ pi 2) ris))
                    (setq ptt2 (polar ptt1 0 tr))
                    (setq p3 (polar ptaa (+ (/ pi 2) (/ pi 4)) (* 1.414 0.009)))
                    (setq p4 (polar ptt1 (/ pi -4) (* 1.414 0.009)))
                    (setq p5 (polar ptt2 (+ (/ pi 2) (/ pi 4)) (* 1.414 0.009)))
                (command "donut" "0.00" 0.018 p3 "")
                (command "pline" ptaa ptt1 "")
                (command "donut" "0.00" 0.018 p4 "")
                (command "pline" ptt1 ptt2 "")
            (setq ptaa ptt2)
            (setq i (+ 1 i))
        )
                (command "pline" ptaa ptt1 "")
                    (setq ptt3 (polar ptt2 0 0.03))
```

```
(command "pline" ptt2 ptt3 "a" (polar ptt3 (/ pi -2) 0.03) (polar
ptt3(+ (/ pi 4) pi) (* 1.414 0.03)) "")

(setq ze3 (atan 0.025 0.10))
(setq zd3 (/ 0.10 (cos ze3)))
(setq ze1 (atan ris tr))
(setq ze2 (/ ze1 2))
(setq zd33 (* (- t 0.025) (/ (sin ze2) (cos ze2))))
(setq zd34 (/ (- t (+ (* 1.414 0.025) 0.025)) (sin ze1)))
(setq pta1 (polar pt6 (- pi ze3) zd3))
(setq pta2 (polar pta1 0 (+ ld zd33)))
(setq pta3 (polar ptt2 0 zd34))
(setq pta4 (polar pta3 0 (+ (- up tr 0.05 zd34) 0.10)))
(setq pta5 (polar pta4 (/ pi -2) 0.30))
(setq pta6 (polar pta4 (+ (/ pi 4) pi) (* 1.414 0.015)))
(setq pta7 (polar pta6 (/ pi -2) 0.20))
(setq pta8 (polar ptt2 (+ (/ pi 4) pi) (* 1.414 0.015)))
(setq pta9 (polar pta8 (+ pi ze1)(+ (/ (* (- num 1) tr) 4) tr)))
(command "pline" pta8 pta6 pta7 "")
(command "pline" pta2 "w" 0.000 "" pta3 pta4 pta5 "")
(command "pline" pta5 "a" (polar pta5 pi 0.03) (polar pta5 (+ (/ pi
2) (/ pi 4)) (* 1.414 0.03)) "")
(command "pline" pta2 pta1 "a" (polar pta1 (/ pi 2) 0.03) (polar
pta1 (/ pi 4) (* 1.414 0.03)) "")
(command "pline" pta8 pta9 "a" (polar pta9 (+ (/ pi -2) ze1) 0.03)
(polar pta9 (+ (/ pi -2) ze1 (/ pi 4)) (* 1.414 0.03)) "")
(command "pline" pta6 pta7 "a" (polar pta7 pi 0.03) (polar pta7
(+ (/ pi 2) (/ pi 4)) (* 1.414 0.03)) "")
(command "pline" ptt1 "w" 0.000 "" ptt1 "")

(setq px1 (polar (polar pt4 (/ pi -2) 0.20) 0 0.10))
(setq px2 (polar (polar pt10 (/ pi -2)(+ hi 0.20)) 0 0.10))
```

```
(command "pline" (polar px1 pi 0.025) px1 px2 (polar px2 0 0.025)"")
(command "pline" (polar px1 (/ pi 2) 0.075) (polar px1 (/ pi -2) 0.025)"")
(command "pline" (polar px2 (/ pi 2) 0.075) (polar px2 (/ pi -2) 0.025)"")

    (setq py1 (polar pt8 0 0.30))
    (setq py2 (polar py1 (/ pi -2) hi))
(command "pline" (polar py1 (/ pi 2) 0.025) py1 py2 (polar py2 (/ pi -2)
0.025)"")
(command "pline" (polar py1 pi 0.20) (polar py1 0 0.025)"")
(command "pline" (polar py2 pi 0.03) (polar py2 0 0.025)"")
(command "pline" (polar (polar py1 pi 0.175)(/ pi 2) 0.025)
        (polar (polar py1 pi 0.175)(/ pi -2)(+ t 0.025))"")
(command "pline" (polar (polar (polar py1 pi 0.175)(/ pi -2) t) pi 0.025)
        (polar (polar (polar py1 pi 0.175)(/ pi -2) t) 0 0.025)"")
    (setq xm (+ ld up (* (- num 1) tr)))
    (setq px3 (polar px1 0 (/ (+ ld up (* (- num 1) tr)) 2)))
    (setq px4 (polar px3 (/ pi -2) 0.08))
    (setq xm (rtos xm 2 2))
 (command "text" px4 0.035 0.00 xm)
 (command "text" (polar (polar (polar py1 pi 0.175)(/ pi -2)(/ t 1.5)) 0
0.04) 0.035 0.00 (rtos t 2 2))
    (setq xm (atof xm))
    (setq px5 (polar px1 (/ pi 2) 0.05))
    (setq px6 (polar px5 0 ld))
    (setq px7 (polar px5 0 (/ ld 2.5)))
    (setq px8 (polar px7 (/ pi 2) 0.025))
    (setq px9 (polar px8 0 (+ (- ld (/ ld 2.5)) (/ (* (- num 1) tr) 2))))
    (setq px10 (polar px9 0 (+ (/ (* (- num 1) tr) 2) (/ up 2))))
    (setq ltr (* (- num 1) tr))
    (setq ld (rtos ld 2 2))
    (setq ltr (rtos ltr 2 2))
    (setq up (rtos up 2 2))
```

```
(command "pline" (polar px5 pi 0.025) (polar (polar px2 0 0.025) (/ pi
2) 0.05)"")
(command "pline" (polar px6 (/ pi 2) 0.025) (polar px6 (/ pi -2) 0.025)"")
(command "pline" (polar (polar px6 0 (* (- num 1) tr)) (/ pi 2) 0.025)
                 (polar (polar px6 0 (* (- num 1) tr)) (/ pi -2)
0.025)"")
(command "text" px8 0.035 0.00 ld)
(command "text" px9 0.035 0.00 ltr)
(command "text" px10 0.035 0.00 up)
    (setq ld (atof ld))
    (setq ltr (atof ltr))
    (setq up (atof up))
    (setq py3 (polar py1 (/ pi -2) (/ hi 2)))
    (setq py4 (polar py3 0 0.06))
    (setq hi (rtos hi 2 2))
(command "text" py4 0.035 0.00 hi)
    (setq hi (atof hi))

    (setq pd1 px4)
    (setq pd2 (polar pd1 (/ pi -2) 0.05))
    (setq pd3 (polar pd2 pi 0.17))
    (setq pd4 (polar pd3 (/ pi -2) 0.08))
    (setq pd5 (polar pd4 0 0.20))
    (setq pd6 (polar pd5 0 0.14))
    (setq pd7 (polar pd6 (/ pi 2) 0.08))
    (setq pd8 (polar pd7 pi 0.14))
    (setq pd9 (polar pd4 0 0.06))
    (setq pd10 (polar pd9 (/ pi 2) 0.015))
    (setq pd11 (polar pd5 0 0.02))
    (setq pd12 (polar pd11 (/ pi 2) 0.015))
(command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5 "")
(command "text" pd10 0.04 0.00 nam)
(command "zoom" "w" pt15 pt16 )
```

```
(command "_pedit" pd2 "w" 0.005 "")

(setq dip (getreal "\nDiameter steel positive moment <12>:"))
(if (not dip)(setq dip 12.00))
(setq spp (getreal "\nSpacing steel <0.20>:"))
(if (not spp)(setq spp 0.20))
(setq dit (getreal "\nDiameter temparature steel <9>:"))
(if (not dit)(setq dit 9.00))
(setq sp (getreal "\nSpacing steel <0.20>:"))
(if (not sp)(setq sp 0.20))
        (setq dit (/ dit 1000))
        (setq rt (/ dit 2))
        (setq ps5 (polar pta1 0 0.15))
        (setq xld (distance ps5 pta2))
        (setq nsl (/ xld sp))
    (if (= nst (fix nsl))
        (setq rol nsl)
        (setq rol (+ (fix nsl) 1))
    )
        (setq spl (/ xld rol))
        (setq ps6 (polar ps5 (/ pi 2) 0.009))
        (setq ps7 (polar ps5 (/ pi 2)(- t 0.05 0.009)))
(setq i 1)
(while (<= i rol)
        (command "donut" "0.00" 0.018 ps6 "")
        (command "donut" "0.00" 0.018 ps7 "")
        (setq ps6 (polar ps6 0 spl))
        (setq ps7 (polar ps7 0 spl))
        (setq i (+ i 1))
)
        (command "donut" "0.00" 0.018 ps6 "");cross section steel lower
landing
```

```
            (setq pv13 (polar pta 0 (+ 0.225 (* spl 2.5))))
            (command "pline" (polar pv13 (- pi (/ pi 3)) 0.025) pv13 (polar pv13
(/ pi 3) 0.025) "")
            (command "pline" pv13 (polar pv13 (/ pi 2) 0.20)(polar (polar pv13
(/ pi 2) 0.20) pi 0.10) "")


              (setq ps8 (polar pt13 pi 0.15))
              (setq xup (- up 0.15 tr))
              (setq nsu (/ xup sp))
          (if (= nsu (fix nsu))
              (setq rou nsu)
              (setq rou (+ (fix nsu) 1))
          )
              (setq spu (/ xup rou))
              (setq ps9 (polar ps8 (/ pi 2) 0.009))
              (setq ps10 (polar ps8 (/ pi 2)(- t 0.05 0.009 0.018)))
    (setq i 1)
    (while (<= i rou)
            (command "donut" "0.00" 0.018 ps9 "")
            (command "donut" "0.00" 0.018 ps10 "")
              (setq ps9 (polar ps9 pi spu))
              (setq ps10 (polar ps10 pi spu))
              (setq i (+ i 1))
     )
            (command "donut" "0.00" 0.018 ps9 "");cross section steel upper
landing


          (setq pv10 (polar pta6 pi (+ (- 0.25 0.04) (* spu 1.5))))
          (setq pv11 (polar (polar pta6 pi (+ (- 0.25 0.04) spu))(/ pi -2) 0.009))
          (setq pv12 (polar pt13 pi (+ 0.15 (* spu 2.5))))
```

```
(command "pline" (polar pv10 (- pi (/ pi 3)) 0.025) pv10 (polar pv10 (/
pi 3) 0.025) "")
(command "pline" pv10 (polar pv10 (/ pi 2) 0.30)(polar (polar pv10 (/
pi 2) 0.30) 0 0.20) "")
(command "pline" (polar pv11 (- pi (/ pi 3)) 0.025) pv11 (polar pv11 (/
pi 3) 0.025) "")
(command "pline" pv11 (polar pv11 (/ pi 2) 0.15)(polar (polar pv11 (/
pi 2) 0.15) 0 0.10) "")
(command "pline" (polar pv12 (+ pi (/ pi 3)) 0.025) pv12 (polar pv12 (/
pi -3) 0.025) "")
(command "pline" pv12 (polar pv12 (/ pi -2) 0.50)(polar (polar pv12 (/
pi -2) 0.50) 0 0.10) "")


        (setq ps11 (polar pta2 (/ pi 2) 0.009))
        (setq ang (angle pta2 pta3))
        (setq xsl (- (car pta3)(car pta2)))
      (if (= ang (/ pi 2))
        (setq xsl (- (cadr pta3)(cadr pta2)))
        (setq xsl (/ xsl (cos ang)))
      )
        (setq ros (/ xsl sp))
       (if (/= ros (fix ros))
        (setq ros (1+ (fix ros)))
       )
        (setq sps (/ xsl ros))
        (setq i 1)
       (while (<= i ros)
        (command "donut" "0.00" 0.018 ps11 "")
        (setq ps11 (polar ps11 ang sps))
        (setq i (1+ i))
       ) ;cross section steel slope lower to upper
      (command "pline" (polar pta2 ze1 (* sps 3.5))
```

```
                    (polar (polar pta2 ze1 (* sps 3.5))(/ pi -2) 0.30)
                    (polar (polar (polar pta2 ze1 (* sps 3.5))(/ pi -2) 0.30) 0
0.20) "")
          (command "pline" (polar (polar pta2 ze1 (* sps 3.5))(+ pi (/ pi 3))
0.025)
                    (polar pta2 ze1 (* sps 3.5))
                    (polar (polar pta2 ze1 (* sps 3.5))(/ pi -3) 0.025) "")
               (setq dip (rtos dip 2 0))
               (setq spp (rtos spp 2 2))
               (setq txo (strcat (strcat "DB" dip "MM @") spp""))
          (command "text" (polar (polar (polar pta2 ze1 (* sps 3.5))(/ pi -2) 0.30)
0 0.225) 0.035 0.0 txo)
          (command "text" (polar (polar pv10 (/ pi 2) 0.30) 0 0.225) 0.035 0.0
txo)
          (command "text" (polar (polar pv12 (/ pi -2) 0.50) 0 0.125) 0.035 0.0
txo)
          (command "text" (polar (polar pv13 (/ pi 2) 0.20) pi 0.50) 0.035 0.0
txo)
               (setq dip (atoi dip))
               (setq spp (atof spp))

               (setq psy (polar pta2 (/ pi 2) 0.009))
          (command "pline" (polar psy ze1 (* sps 5))
                    (polar (polar psy ze1 (* sps 5))(/ pi -2) 0.30)
                    (polar (polar (polar psy ze1 (* sps 5))(/ pi -2) 0.30) 0 0.20)
"")
          (command "pline" (polar (polar psy ze1 (* sps 5))(+ pi (/ pi 3)) 0.025)
                    (polar psy ze1 (* sps 5))
                    (polar (polar psy ze1 (* sps 5))(/ pi -3) 0.025) "")
               (setq dit (* dit 1000))
               (setq dit (rtos dit 2 0))
               (setq sp (rtos sp 2 2))
               (setq txt (strcat (strcat "RB" dit "MM @") sp ""))
```

```
        (command "text" (polar (polar (polar psy ze1 (* sps 5))(/ pi -2) 0.30) 0
0.225) 0.035 0.0 txt)
        (command "text" (polar (polar pv8 (/ pi 2) 0.30) pi (+ 0.525 (* tr 1.5)))
0.035 0.0 txt)
        (command "text" (polar (polar pv11 (/ pi 2) 0.15) 0 0.125) 0.035 0.0
txt)

            (setq dit (atoi dit))
            (setq sp (atof sp))
            (setq dit (rtos dit 2 0))
            (setq txt (strcat (strcat "1-RB" dit "MM") ""))
        (command "text" (polar (polar pv9 (/ pi 2) 0.15) pi (+ 0.40 tr 0.025
0.009)) 0.035 0.0 txt)
            (setq dit (atoi dit))

              (setq ps12 (polar pta9 ze1 0.025))
              (setq ps13 (polar ps12 (+ (/ pi -2) ze1) 0.009))
              (setq ang (angle pta9 pta8))
              (setq xs (- (car pta8)(car pta9)))
           (if (= ang (/ pi 2))
              (setq xs (- (cadr pta8)(cadr pta9)))
              (setq xs (/ xs (cos ang)))
           )
              (setq ros (/ xs sp))
            (if (/= ros (fix ros))
              (setq ros (1+ (fix ros)))
            )
              (setq sps (/ xs ros))
              (setq i 1)
           (while (<= i ros)
              (command "donut" "0.00" 0.018 ps13 "")
              (setq ps13 (polar ps13 ang sps))
              (setq i (1+ i))
```

```
);cross section steel at L/4 from upper landing

1.          (command "zoom"  "p")
             (initget 2 " 1 2 3 ")
                (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3 :"))
                (setq sc (atoi sca))
           (if (> sc 2)
                  (command "text" pd12 0.04 0.00 "1:25")
              (if (< sc 2)
                  (command "text" pd12 0.04 0.00 "1:10")
                  (command "text" pd12 0.04 0.00 "1:20")
              )
            )
              (cond
                  (( = sca "1") (command "scale" "w" pt16 pt15 "" bm 10))
                  (( = sca "2") (command "scale" "w" pt16 pt15 "" bm 5))
                  (( = sca "3") (command "scale" "w" pt16 pt15 "" bm 4))
              )
            (command "zoom" "a")
            (princ)
            (princ)
          ) ;end stairs
```

```
(vmon)
(defun c:stair1l ()
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq bm (getpoint "\nSelect base point:"))
(setq hi (getreal "\nFloor to Floor Height <1.50>:"))
(if (not hi)
(setq hi 1.50)
)
(setq ris (getreal "\nRiser <0.175>:"))
(if (not ris)
(setq ris 0.175)
)
(setq tr (getreal "\nTread <0.25>:"))
(if (not tr)
(setq tr 0.25)
)
(setq t1 (getreal "\nThickness stair<0.12>:"))
(if (not t1)
(setq t1 0.12)
)
(setq ld (getreal "\nWidth Landing low <1.20> :"))
(if (not ld)
(setq ld 1.20)
)
(setq up (getreal "\nWidth Landing up <1.20>:"))
(if (not up)
(setq up 1.20)
)
(setq ze1 (atan ris tr))
(setq ze2 (/ ze1 2))
(setq zd (/ t1 (cos ze2)))
(setq pt01 (polar bm pi tr))
```

```
(setq pt3 (polar bm pi ld))
(setq pt4 (polar pt3 (/ pi -2) 0.40))
(setq pt5 (polar pt4 0 0.20))
(setq pt6 (polar pt5 (/ pi 2) (- 0.40 t1)))
(setq pt7 (polar pt01 (- (+ pi (/ pi 2)) ze2) zd))
(command "pline" bm pt3 pt4 pt5 pt6 pt7 "")
(setq num (/ hi ris))
(if (/= num (fix num))
(setq num (+ 1 (fix num)))
)
(setq ris (/ hi num))
(setq i 1)
(while (<= i num)
(setq p1 (car bm))
(setq p2 (cadr bm))
(setq pt1 (list p1 (- p2 ris)))
(setq pt2 (list (+ p1 tr) (- p2 ris)))
(command "pline" bm pt1 pt2 "")
(setq bm pt2)
(setq i (+ 1 i))
)
(command "pline" bm pt1 "")
(setq pt0 (polar pt1 0 tr))
(setq pt8 (polar pt1 0 up))
(setq pt9 (polar pt8 (/ pi -2) 0.40))
(setq pt10 (polar pt9 pi 0.20))
(setq pt11 (polar pt10 (/ pi 2) (- 0.40 t1)))
(setq pt12 (polar pt1 (- (+ pi (/ pi 2)) ze2) zd))
(command "pline" pt1 pt8 pt9 pt10 pt11 pt12 pt7 "")
)
```

COLUMNS.LSP

```
                    (vmon)
(defun c:columns()
                    (setvar "BLIPMODE" 0)
                    (setvar "CMDECHO" 0)
                        (setq nam (getstring "\nName of column:"))
                        (setq stp (getpoint "\nStart point at low left corner:"))

                        (setq wc (getdist stp "\nWidth of column<0.30>:"))
                    (if (not wc)
                        (setq wc 0.30)
                    )
                        (setq hc (getdist stp "\nHeight of column<0.30>:"))
                    (if (not hc)
                        (setq hc 0.30)
                    )
                        (setq co (getreal "\nConcrete covering <0.03>:"))
                        (if (not co) (setq co 0.03))
                    (while (or (< co 0.03)(> co 0.05))
                        (setq co (getreal "\nConcrete covering <0.03>:"))
                        (if (not co) (setq co 0.03))
                    )

                        (setq pt1 (polar stp (/ pi 2) hc))
                        (setq pt2 (polar pt1 0 wc))
                        (setq pt3 (polar pt2 (/ pi -2) hc))
                        (setq pt4 (polar stp pi 0.07))
                        (setq pt5 (polar pt1 pi 0.07))
                        (setq pt6 (polar pt1 (/ pi 2) 0.07))
                        (setq pt7 (polar pt2 (/ pi 2) 0.07))
                    (command "pline" stp "w" 0.00 stp"")
                    (command "pline" stp pt1 pt2 pt3 "c")
```

```
                (command "offset" co stp (polar stp (/ pi 4) (* 1.414 co)) "")
                (command "_pedit" (polar (polar stp (/ pi 4) (* 1.414 co))  (/ pi
2) 0.025) "w" 0.005"")
                (command "pline" pt4 pt5 ""  "pline" pt6 pt7 "")
                (command "pline" (polar pt4 0 -0.025) (polar pt4 0 0.025) "")
                (command "pline" (polar pt5 0 -0.025) (polar pt5 0 0.025) "")
                (command "pline" (polar pt6 (/ pi 2) -0.025) (polar pt6 (/ pi 2)
0.025) "")
                (command "pline" (polar pt7 (/ pi 2) -0.025) (polar pt7 (/ pi 2)
0.025) "")
                    (setq pyy (polar pt4 (/ pi 2) (/ hc 3)))
                    (setq py (polar pyy pi 0.05))
                    (setq hc (rtos hc 2 2))
                (command "text" py 0.04 90 hc )
                   (setq hc (atof hc))
                    (setq pxx (polar pt6 0 (/ wc 3.5)))
                    (setq px (polar pxx (/ pi 2) 0.03))
                    (setq wc (rtos wc 2 2))
                (command "text" px 0.04 0.00 wc )
                    (setq wc (atof wc))
                    (setq pt12 (polar pt1 (/ pi 1.3) (* wc 2.5)))
                    (setq pt13 (polar pt3 (/ pi -6) (* wc 5)))
                    (setq pd1 (polar stp 0 (/ wc 2)))
                    (setq pd2 (polar pd1 (/ pi -2) 0.07))
                    (setq pd3 (polar pd2 pi 0.17 ))
                    (setq pd4 (polar pd3 (/ pi -2) 0.08))
                    (setq pd5 (polar pd4 0 0.20))
                    (setq pd6 (polar pd5 0 0.14))
                    (setq pd7 (polar pd6 (/ pi 2) 0.08))
                    (setq pd8 (polar pd7 pi 0.14))
                (command "pline" pd8 pd3 pd4 pd6 pd7 pd8 pd5"")
                    (setq pd9 (polar pd4 0 0.06))
                    (setq pd10 (polar pd9 (/ pi 2) 0.015))
```

```
(command "text" pd10 0.04 0.00 nam)
(command "zoom"  "w" pt12 pt13 )
(command "_pedit" pd2  "w" 0.005"")
    (setq ns1 (getint "\nNumber of stirrup steel<1> :"))
(if (not ns1)(setq ns1 1 ))

    (setq ds1 (getreal "\nDiameter stirrup steel<6>:"))
(if (not ds1)(setq ds1 6 ))
(if (< ds1 12.0)(setq ch1 "RB")(setq ch1 "DB"))
    (setq sp1 (getreal "\nSpacing stirrup steel<0.20>:"))
 (if (not sp1)
   (setq sp1 0.20)
 )
    (setq css (itoa ns1))
    (setq dss (* ds1 1))
    (setq sp (rtos sp1 2 2))
    (setq ds (rtos dss 2 0))
    (setq txo (strcat (strcat (substr css 1 1) ch1 ds"MM @")sp""
))
   (command "text" (polar pt7 0 0.10) 0.035 0.0 txo )
(defun steel (pt1 ang1 ang2 tx)
    (if (< dis 12.0)(setq ch2 "RB")(setq ch2 "DB"))
    (setq rs (/ (/ dis 2) 1000))
    (setq dis (* rs 2))
    (setq rco (+ co (+ rs 0.005)))

    (setq spc (- (/ (- hc (* 2 rco)) (- row 1)) dis))
 (while (< spc 0.025)
   (progn
    (setq row (getreal "\nNumber of layer steel :"))
    (setq spc (- (/ (- hc (* 2 rco)) (- row 1)) dis))
```

```
                (setq ps1 (polar pt1 0.0 (- wc rco)))
                (setq ps2 (polar ps1 (/ pi ang1) (+ rco y)))
                (setq dxc (/ (- wc (* 2 rco)) (- nus 1)))
        (if (> (- dxc (* rs 2)) 0.025)
          (progn
                (setq x 0)
            (repeat nus
                (setq ps3 (polar ps2 pi x))
                (setq x (+ x dxc))
              (command "donut" "0.0" dis ps3 "")
                (setq ps4 (polar ps3 (/ pi -2) (/ 0.035 2)))
                (setq ps5 (polar ps4 0 (- (+ wc 0.10) rco)))
            )
                (setq cn (itoa nus))
                (setq dis (* dis 1000))
                (setq cs (rtos dis 2 0))
                (setq txt (strcat (substr cn 1 1) ch2 cs "MM"))
                (setq ptx ps5)
              (command "text" ptx 0.035 0.0 txt)
                (setq y (+ y (+ spc  (* 2 rs))))
          );progn
        );if
);end steel

            (setq row (getreal "\nNumber of layer steel :"))

            (setq y 0.0)
            (setq nus 1)
        (while nus
          (progn
                (setq nus (getint "\nNumber of lower steel :"))
            (if nus
              (progn
```

```
                    (setq dis (getreal "\nDiameter steel <mm>:"))
                    (steel stp 2 3 0.04)
                )
            )
        )
    )

        (setq y 0.0)
        (setq nus 1)
    (while nus
        (progn
            (setq nus (getint "\nNumber of upper steel :"))
            (if nus
                (progn
                    (setq dis (getreal "\nDiameter steel< mm >:"))
                    (steel pt1 -2 -3 0.04)
                )
            )
        )
    )
            (command "zoom"  "p" )
                (setq pd11 (polar pd5 0 0.02))
                (setq pd12 (polar pd11 (/ pi 2) 0.015))
      (initget 2 " 1 2 3 ")
                (setq sca (getkword "\nScale1:10=1,1:20=2,1:25=3
:"))
                (setq sc (atoi sca))
            (if (> sc 2)
                (command "text" pd12 0.04 0.00 "1:25")
                (if (< sc 2)
                (command "text" pd12 0.04 0.00 "1:10")
                (command "text" pd12 0.04 0.00 "1:20")
```

```
                    )
                    (cond
                        (( = sca "1") (command "scale"  "w" pt12 pt13 "" stp 10))
                        (( = sca "2") (command "scale"  "w" pt12 pt13 "" stp 5))
                        (( = sca "3") (command "scale"  "w" pt12 pt13 "" stp 4))
                    )
(princ)
(princ)
) ;end columns
        (SETVAR "BLIPMODE" 1)
        (SETVAR "CMDECHO" 1)
```

FOOTINGS.LSP

```
(vmon)
(defun c:footings (/ name bm x1 x2 LS wc wcz y1 y2 co y3 y4 y5
pt1 pt2 pt3 pt4 pt5 pt6 pt7 pt8 pt9 pt10 pt11
pt12 pt13 pt14 pt15 px1 px2 px3 px4 px5 px6 px7
px8 px9 py1 py2 py4 py4 py5 py6 py7
py8 pd1 pd2 pd3 pd4 pd5 pd6 pd7 pd8 pd9 pd10
pd11 ps1 ps2 ps3 ps4 ps5 ps6 ps7 ps8
ps9 x psl psr psss pss)
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq name (getstring "\nName of footing :"))
(setq bm (getpoint "\nFirst point lower center of
footing :"))
(setq x1 (getdist bm "\nLeft cantilever ..x
direction..<0.50>:"))
(if (not x1)(setq x1 0.50))
(setq x2 (getdist bm "\nRight cantilever ..x
direction..<1.00>:"))
(if (not x2)(setq x2 1.00))
(setq LS (getdist bm "\nLength of short span ..y
direction..<0.80>:"))
(if (not LS)(setq LS 0.80))
(setq wc (getreal "\nSize of column ..x
direction..<0.25> :"))
(if (not wc)(setq wc 0.25))
(setq wcz (getreal "\nSize of column ..y direction..<0.20> :"))
(if (not wcz)(setq wcz 0.20))
(setq y1 (getdist bm "\nThickness of footing <0.40>
:"))
(if (not y1)(setq y1 0.40))
(setq y2 (getreal "\nSlope footing <0.00> :"))
```

```
                    (if (not y2)(setq y2 0.00))
                (setq co (getreal  "\nConcrete covering <0.05> :"))
1.                  (if (not co)(setq co 0.05))
                (setq y3 (getdist bm  "\nDepth of footing <1.20> :"))
                    (if (not y3)(setq y3 1.20))
                (setq y4 (getdist bm  "\nThickness of lean concrete
<0.10> :"))
                    (if (not y4)(setq y4 0.10))
                (setq y5 (getdist bm  "\nThickness of sand <0.10>
:"))
                    (if (not y5)(setq y5 0.10))

            (if (= x1 0.00)
                (setq x1 (/ wc 2))
            )
                (if (= x2 0.00)
                  (setq x2 (/ wc 2))
                )
                  (setq pt1 (polar bm pi x1))
                  (setq pt2 (polar pt1 (/ pi 2) y1))
                  (setq pt3 (list (+ (car pt1) (- x1 (/ wc 2))) (+ (cadr
pt1) (+ y1 y2))))
                  (setq pt4 (polar pt3 (/ pi 2) (- y3 (+ y1 y2))))
                (if (> (+ y1 y2) y3)
                  (setq pt5 (polar pt3 (/ PI 2) 0.25))
                  (setq pt5 (polar pt4 (/ pi 2) 0.25))
                )
                  (setq pt6 (polar pt4 pi (+ (- x1 (/ wc 2)) 0.15)))
                  (setq pt7 (polar bm 0 x2))
                  (setq pt8 (polar pt7 (/ pi 2) y1))
                  (setq pt9 (list (+ (car pt3) wc) (cadr pt3)))
                  (setq pt10 (polar pt5 0 wc))
                  (setq pt11 (polar pt5 0 (- (/ wc 2) 0.035)))
```

```
                    (setq pt12 (polar pt10 pi (- (/ wc 2) 0.035)))
                (command "pline" pt5 pt3 pt2 pt1 pt7 pt8 pt9 pt10"")
                    (if (> (+ y1 y2) y3)
                        (command "pline" (polar pt4 pi (- (+ x1 0.05) (/ wc
2))) (polar pt4 0 (+ (+ wc x2) 0.20))"")
                        (command  "pline" pt6 pt4 "")
                    )
                        (command  "pline" (polar pt4 0 wc) (polar pt4 0 (+
wc x2))""
                                "pline" (polar pt1 (/ pi -2) y4) (polar pt7 (/ pi
-2) y4)""
                                "pline" (polar pt1 (/ pi -2) (+ y4 y5)) (polar
pt7 (/ pi -2) (+ y4 y5))"")

    (command "text"   (polar (polar pt4 0 (+ wc 0.10)) (/ pi 2) 0.025) 0.035
0.00 "Ground Level")


    (command "pline" (polar (polar bm (/ pi -2) (/ y4 2)) (+ pi (/ pi 3))  (* 1.414
0.025))  (polar bm (/ pi -2) (/ y4 2))
                        (polar (polar bm (/ pi -2) (/ y4 2)) (/ pi -2) (+ (+ (/ y4
2) y5) 0.075))
                        (polar (polar (polar bm (/ pi -2) (/ y4 2)) (/ pi -2) (+
(+ (/ y4 2) y5) 0.075)) 0 0.15)""
                "text" (polar (polar (polar (polar bm (/ pi -2) (/ y4 2)) (/ pi -2)
(+ (+ (/ y4 2) y5) 0.075)) 0 0.15) 0 0.025)
                        0.035 0.00 "Lean concrete 1:3:5 "
                "pline" (polar (polar (polar bm pi 0.10) (/ pi -2) (+ (/ y5 2) y4))
(+ pi (/ pi 3)) (* 1.414 0.025))
                        (polar (polar bm pi 0.10) (/ pi -2) (+ (/ y5 2) y4))
                        (polar (polar (polar bm pi 0.10) (/ pi -2) (+ (/ y5 2)
y4)) (/ pi -2) (+ (/ y5 2) 0.15))
```

```
(polar (polar (polar (polar bm pi 0.10) (/ pi -2) (+ (/ y5 2) y4)) (/
pi -2) (+ (/ y5 2) 0.15)) 0 0.25)""
"text" (polar (polar (polar (polar (polar bm pi 0.10) (/ pi -2) (+
(/ y5 2) y4)) (/ pi -2) (+ (/ y5 2) 0.15)) 0 0.25) 0 0.025)
0.035 0.00 "Sand compaction ")

(command "pline" (polar pt5 pi 0.025)  pt11 (polar pt11 (/ pi 2) 0.025)
(polar pt12 (/ pi -2) 0.025)
pt12 (polar pt10 0 0.025)""
"pline" (polar (polar pt5 (/ pi 2) 0.10) pi 0.025) (polar (polar
pt10 (/ pi 2) 0.10) 0 0.025)""
"pline" (polar (polar pt5 (/ pi 2) 0.10) (/ pi 2) 0.025) (polar
(polar pt5 (/ pi 2) 0.10) (/ pi -2) 0.025)""
"pline" (polar (polar pt10 (/ pi 2) 0.10) (/ pi 2) 0.025) (polar
(polar pt10 (/ pi 2) 0.10) (/ pi -2) 0.025)""
"text" (polar  (polar (polar pt5 (/ pi 2) 0.10) (/ pi 2) 0.025)  0
(/ wc 3)) 0.035 0.00 (rtos wc 2 2))

(setq x (+ x1 x2))
(setq px1 (polar pt1 (/ pi -2) (+ (+ y4 y5) 0.35)))
(setq px2 (polar px1 0 x))
(setq px3 (polar px1 0 x1))
(command "pline" (polar px1 pi 0.025) px1 px2 (polar px2
0 0.025)""
"pline" (polar px1 (/ pi 2) 0.10) (polar px1 (/
pi -2) 0.025)""
"pline" (polar px2 (/ pi 2) 0.10) (polar px2 (/
pi -2) 0.025)""
"text" (polar px3 (/ pi -2) 0.08) 0.035 0.00
(rtos x 2 2))
(setq px4 (polar px1 (/ pi 2) 0.075))
(setq px5 (polar px4 0 x1))
(setq px6 (polar px4 0 (/ x1 2.5)))
```

```
                    (setq px7 (polar px6 (/ pi 2) 0.025))

                (command "pline" (polar px4 pi 0.025) (polar px4 0 (+ x
0.025))""
                    "pline" (polar px5 (/ pi 2) 0.025) (polar px5 (/
pi -2) 0.025)""
                    "text" px7 0.035 0.00 (rtos x1 2 2)
                "text"  (polar px7 0 (- (- x (/ x1 2)) (/ X2 2))) 0.035
0.00 (rtos x2 2 2))

                (setq py1 (polar pt6 pi 0.10))
                (setq py2 (polar py1 (/ pi -2) y3))
                (setq py3 (polar py1 (/ pi -2) (/ y3 2)))
            (command "pline" (polar py1 (/ pi 2) 0.025) py1 py2 (polar
py2 (/ pi -2) 0.025)""
                    "pline" (polar py1 pi 0.025) (polar py1 0
0.025)""
                    "pline" (polar py2 pi 0.025) (polar py2 0
0.175)""
                    "text" (polar py3 pi 0.10) 0.035 0.00 (rtos y3
2 2))
                (setq py4 (polar pt1 pi 0.10))
                (setq py5 (polar py4 (/ pi 2) y1))
                (setq py6 (polar py5 (/ pi 2) y2))
                (setq py7 (polar py5 (/ pi 2) (/ y2 2)))
                (setq py8 (polar py4 (/ pi 2) (/ y1 2)))
            (command "pline" (polar py6 (/ pi 2) 0.025) (polar py4 (/ pi
-2) (+ (+ y4 y5) 0.025))""
                    "pline" (polar py6 pi 0.025) (polar py6 0
0.025)""
                    "pline" (polar py5 pi 0.025) (polar py5 0
0.025)""
```

```
                                        "pline" (polar (polar pt1 (/ pi -2) y4) pi 0.125)
(polar (polar pt1 (/ pi -2) y4) pi 0.075)""
                                        "pline" (polar (polar pt1 (/ pi -2) (+ y4 y5)) pi
0.125)  (polar (polar pt1 (/ pi -2) (+ y4 y5)) pi 0.075)"")
                (if (/= y2 0.00)
                    (command "text" (polar py7 pi 0.10) 0.035 0.00 (rtos y2 2
2))
                )
                    (command "text" (polar py8 pi 0.10) 0.035 0.00 (rtos y1 2
2))
            (command "text" (polar (polar py8 pi 0.10)  (/ pi -2) (+ (/ y1 2) (/ y4
1.3)))  0.035 0.00 (rtos y4 2 2)
                        "text" (polar (polar py8 pi 0.10)  (/ pi -2) (+ (+ (/ y1 2) (/
y5 1.3)) y4))  0.03 0.00 (rtos y5 2 2))

                            (setq pd1 (polar px3 (/ pi -2) 0.08))
                            (setq pd2 (polar pd1 (/ pi -2) 0.05))
                            (setq pd3 (polar pd2 pi 0.17))
                            (setq pd4 (polar pd3 (/ pi -2) 0.08))
                            (setq pd5 (polar pd4 0 0.34))
                            (setq pd6 (polar pd5 (/ pi 2) 0.08))
                            (setq pd7 (polar pd6 pi 0.14))
                            (setq pd8 (polar pd4 0 0.06))
                            (setq pd9 (polar pd8 (/ pi 2) 0.015))
                            (setq pd10 (polar pd4 0 0.22))
                            (setq pd11 (polar pd10 (/ pi 2) 0.015))
                        (command "pline" pd7 pd3 pd4 pd5 pd6 pd7 (polar pd4 0
0.20)"")
                    (command "text" pd9 0.04 0.00 name)
                (if (> (+ y1 y2) y3)
                        (setq pt13 (polar pt5 (- pi (/ pi 5)) (+ x1 0.60)))
                        (setq pt13 (polar pt6 (- pi (/ pi 3)) 0.60))
                )
```

```
                    (setq pt14 (polar px2 (/ pi -4) 0.60))
                (command "zoom" "w" pt13 pt14)
                (command "_pedit" pd2 "w" 0.005"")

                (initget 2 " B ")
            (setq sele (getkword "\nSelect steel long span  A ,B  <A> :"))
(cond
  (( = sele "B")
  (command "pline" (polar (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) 0.03) 0
0.03)
                    (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) 0.03)  "a"
(polar pt1(/ pi 4) (* 1.414 co)) "l"
                    (polar pt7(- pi (/ pi 4)) (* 1.414 co)) "a" (polar (polar
pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2) 0.035) "l"
                    (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.035)  pi 0.03)""
            "_pedit" (polar pt1 (/ pi 4) (* 1.414 co)) "w" 0.005"")
    (command "pline" (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.035) (- pi (/ pi 3)) (* 1.414 0.025))
                    (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.035)
                    (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.035) (/ pi 2) (+ y1 0.05))
                    (polar (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co))
(/ pi 2) 0.035) (/ pi 2) (+ y1 0.05)) 0 0.10)"")

  ) ; sele
  ((null sele)
  (command "pline" (polar (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) (- y1 (+
co 0.05))) 0 0.10)
                    (poiar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) (- y1 (+ co
0.05)))      (polar pt1(/ pi 4) (* 1.414 co))
```

```
                    (polar pt7(- pi (/ pi 4)) (* 1.414 co))  (polar (polar pt7(-
pi (/ pi 4)) (* 1.414 co)) (/ pi 2) (- y1 (+ co 0.05)))
                    (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
(- y1 (+ co 0.05)))  pi 0.10)""
                "_pedit" (polar pt1 (/ pi 4) (* 1.414 co)) "w" 0.005"")
          (command "_fillet" "R" 0.035)
          (command "_fillet" (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) 0.025)
(polar (polar pt1(/ pi 4) (* 1.414 co)) 0 0.055))
          (command "_fillet" (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) 0.025)
(polar (polar (polar pt1(/ pi 4) (* 1.414 co)) (/ pi 2) (- y1 (+ co 0.05))) 0 0.025))
          (command "_fillet" (polar (polar pt7 (- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.025) (polar (polar pt7 (- pi (/ pi 4)) (* 1.414 co)) pi 0.055))
          (command "_fillet" (polar (polar pt7 (- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)
0.05) (polar (polar (polar pt7 (- pi (/ pi 4)) (* 1.414 co)) (/ pi 2)(- y1(+ co 0.05))) pi
0.05))
   (command "pline"  (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2) (-
y1 (+ co 0.06))) (- pi (/ pi 3)) (* 1.414 0.025))
                    (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi 2) (- y1
(+ co 0.06)))
                    (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi
2) (- y1 (+ co 0.06))) (/ pi 2) (+ co 0.13))
                    (polar (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co))
(/ pi 2) (- y1 (+ co 0.06))) (/ pi 2) (+ co 0.13)) 0 0.10)"")
                    (setq diar (getreal  "\nDiameter steel to around footing
<12> :"))
                    (if (not diar)(setq diar 12.0))
                    (if (< diar 12.0)(setq chr "RB")(setq chr "DB"))
                    (setq diar (/ diar 1000))
                    (setq rar (/ diar 2))
   (command "circle" (polar  (polar (polar pt1 0 (+ co 0.03)) (/ pi 2) (+ (+ co rar)
(/ rar 2))) (/ pi 2) (- y1 (+ (+ co 0.07) diar)))  rar)
   (command "circle" (polar  (polar (polar pt1 0 (+ co 0.03)) (/ pi 2) (+ (+ co rar)
(/ rar 2))) (/ pi 2) (- y1 (+ (+ co 0.07) diar)))  (+ rar 0.007))
```

```
(command "circle" (polar  (polar (polar pt7 pi (+ co 0.03)) (/ pi 2) (+ (+ co rar)
(/ rar 2))) (/ pi 2) (- y1 (+ (+ co 0.07) diar))) rar)
(command "pline" (polar (polar pt2 0 (+ co 0.03)) (/ pi -2) (+ diar 0.05))
                 (polar (polar (polar pt2 0 (+ co 0.03)) (/ pi -2) (+ diar
0.05)) (/ pi 2) (+ (+ diar 0.05) 0.30))
                 (polar (polar (polar (polar pt2 0 (+ co 0.03)) (/ pi -2) (+
diar 0.05)) (/ pi 2) (+ (+ diar 0.05) 0.30)) pi 0.10)"")
                 (setq diar (* diar 1000))
                 (setq diar (rtos diar 2 0))
                 (setq nr 1)
                 (setq nr (rtos nr 2 0 ))
          (setq txr (strcat (substr nr 1 2) chr diar "mm"))
(command "text"  (polar (polar (polar (polar pt2 0 (+ co 0.03)) (/ pi -2) (+
0.012 0.05)) (/ pi 2) (+ (+ 0.012 0.05) 0.30)) pi 0.30)  0.035 0.0 txr)

) ;null
) ;cond

                        (setq dial (getreal  "\nDiameter steel to long
span <16> :"))
                    (if (not dial)(setq dial 16))
                    (if (< dial 12)(setq ch1 "RB")(setq ch1 "DB"))
                        (setq nl (getreal  "\nNumber steel <8> :"))
                    (if (not nl) (setq nl 8))
                        (setq dial (rtos dial 2 0))
                 (setq nl (rtos nl 2 0 ))
               (setq txl (strcat (substr nl 1 2) ch1 dial "mm"))
(command "text" (polar (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co)) (/ pi
2) 0.03) (/ pi 2) (+ y1 0.05)) 0 0.125) 0.035 0.0 txl)

            (setq dias (getreal "\nDiameter steel to short span <12> :"))
        (if (not dias) (setq dias 12.0))
        (if (< dias 12)(setq ch2 "RB")(setq ch2 "DB"))
```

```
                (setq ns (getreal "\nNumber steel <7> :"))
            (if (not ns) (setq ns 7))
                (setq dias (/ dias 1000))
                (setq ras (/ dias 2))
                (setq psl (polar (polar pt1 0 (+ co 0.03)) (/ pi 2) (+ (+ co ras) (/
ras 2))))
                (setq psr (polar (polar pt7 pi (+ co 0.03)) (/ pi 2) (+ (+ co ras) (/
ras 2))))
                (setq xlf (- (car psr)(car psl)))
                (setq ro (- ns 1))
              (if (/= ro (fix ro))
                (setq ro (1+ (fix ro)))
              )
                (setq sp (/ xlf ro))
                (setq i 1)
              (while (<= i ro)
                (command "circle" psl ras)
                (setq psl (polar psl 0 sp))
                (setq i (1+ i))
              )
                (command "circle" psl ras)
                (command "circle" psr (+ ras 0.007))
   (command "pline" (polar psr (/ pi 1.3) (+ ras 0.007))  (polar (polar psr (/ pi 1.3)
(+ ras 0.007)) (- pi (/ pi 3)) 0.05)
                        (polar (polar (polar psr (/ pi 1.3) (+ ras 0.007)) (- pi (/
pi 3)) 0.05) (/ pi 2) (+ y1 0.10))
                        (polar (polar (polar (polar psr (/ pi 1.3) (+ ras 0.007)) (-
pi (/ pi 3)) 0.05) (/ pi 2) (+ y1 0.10)) 0 (+ 0.06 0.10))"")
                                (setq dias (* dias 1000))
                                (setq dias (rtos dias 2 0))
                        (setq ns (rtos ns 2 0 ))
                      (setq txs (strcat (substr ns 1 2) ch2 dias "mm"))
```

```
(command "text" (polar (polar (polar (polar (polar pt7(- pi (/ pi 4)) (* 1.414 co))
(/ pi 2) 0.03) (/ pi 2) (+ y1 0.04)) 0 0.125)
                        (/ pi 2) 0.10) 0.035 0.0 txs)
                            (setq dias (atof dias))
                                (setq dias (/ dias 1000))

                                (setq ps1 (polar bm pi (- (/ wc 2) 0.025)))
                                (setq ps2 (polar ps1 (/ pi 2) (+ (+ co dias) (/
dias 1.5))))
                                (setq ps3 (polar pt5 0 0.025))
                                (setq ps4 (polar ps2 pi (/  x1 2)))
                                (setq ps5 (polar ps3 (/ pi -2) 0.05))
                                (setq ps6 (polar ps2 (/ pi 2) 0.10))
                                (setq ps7 (polar ps6 0 (- wc 0.05)))

                                (setq his (- (car ps7) (car ps6)))

                            (setq dcol (getreal "\nDiameter steel of column <12>
:"))
        (if (not dcol) (setq dcol 12.0))
        (if (< dcol 12)(setq ch3 "RB")(setq ch3 "DB"))
            (setq ncol (getreal "\nNumber steel side view of column <2>
:"))
        (if (not ncol) (setq ncol 2))
            (setq dcol (/ dcol 1000))
            (setq rcol (/ dcol 2))
            (setq hcol (distance ps2 ps3))
            (setq xcol (distance ps6 ps7))
            (setq rcol (- ncol 1))
            (setq spcl (/ xcol rcol))
            (setq ps8 (polar ps2 0 spcl))
            (setq i 1)
          (while (< i rcol)
```

```
                    (setq ps9 (polar ps8 (/ pi 2) hcol))
                    (command "pline" ps8 ps9"")
1.                  (command "_pedit" (polar ps8 (/ pi 2) 0.05) "w" 0.005 "")
                    (setq ps8 (polar ps8 0 spcl))
                    (setq i (1+ i))
                )

            (if (= x1 (/ wc 2))
                (progn
                    (setq pss (polar ps1 (/ pi 2) (+ (+ co (* dias 3)))))
                    (setq psss (polar pss 0 (+ (/ x2 3) wc)))
                    (command "pline"  ps3 pss psss "a" (polar psss (/ pi 2)
0.03) "l" (polar psss (- pi (/ pi 4)) (* 1.414 0.03))"")
                    (command "_pedit" (polar psss (/ pi 2) 0.03) "w" 0.005"")
                    (command "_fillet" "R" 0.05)
                    (command "_fillet" (polar pss 0 0.05) (polar ps2 (/ pi 2)
0.15))
                )
                    (progn
                        (command "pline"  ps3 ps2 ps4 "a" (polar ps4 (/ pi 2) 0.03)
"l" (polar ps4 (/ pi 4) (* 1.414 0.03))"")
                        (command "_pedit"  (polar ps4 (/ pi 4) (* 1.414 0.03))  "w"
0.005"")
                        (command "_fillet" "R" 0.05)
                        (command "_fillet" (polar ps2 pi 0.05) (polar ps2 (/ pi 2)
0.05))
                    )
            )
            (if (= x2 (/ wc 2))
                (progn
                    (setq pso (polar ps1 0 (- wc 0.05)))
                    (setq psp (polar pso (/ pi 2) (+ (+ co (* dias 3)))))
```

```
                    (command "pline" (polar pt10 pi 0.025) psp (polar psp pi (+
(/ x1 3) wc)) "a"
                                (polar (polar psp pi (+ (/ x1 3) wc)) (/ pi 2) 0.03)
"l"
                                (polar (polar (polar psp pi (+ (/ x1 3) wc)) (/ pi 2)
0.03) 0 0.03)"")
                    (command "_pedit"  (polar (polar (polar psp pi (+ (/ x1 3)
wc)) (/ pi 2) 0.03) 0 0.03) "w" 0.005"")
                    (command "_fillet" "R" 0.05)
                    (command "_fillet" (polar psp pi 0.05) (polar psp (/ pi 2)
0.15))
                )
                (progn
                    (command "pline" (polar pt10 pi 0.025) (polar (polar pt10 pi
0.025) (/ pi -2) (distance ps3 ps2))
                                (polar (polar ps2 0 (- wc 0.05)) 0 (/
x2 2)) "a"
                                (polar (polar (polar ps2 0 (- wc 0.05)) 0 (/ x2 2))  (/
pi 2) 0.03) "l"
                                (polar (polar (polar (polar ps2 0 (- wc 0.05)) 0 (/ x2
2))  (/ pi 2) 0.03) pi 0.03)"")
                    (command "_pedit" (polar (polar (polar (polar ps2 0 (- wc
0.05)) 0 (/ x2 2))  (/ pi 2) 0.03) pi 0.03) "w" 0.005"")
                    (command "_fillet" "R" 0.05)
                    (command "_fillet" (polar ps7 (/ pi -2) 0.05) (polar ps2 0 wc))
                )
            );if

                                (setq sp1 (getreal "\nSpacing steel stirrup
column<0.20> :"))
                        (if (not sp1)
                                (setq sp1 0.20)
                        )
```

```
                    (if (> (+ y1 y2) y3)
                        (setq xss (- (+ (+ y1 y2) 0.30) (+ (+ (+ co dias) (/
dias 2)) 0.15)))
                        (setq xss (- (+ y3 0.30) (+ (+ (+ co dias) (/ dias 2))
0.15)))
                    )
                        (setq ros (/ xss sp1))
                (if (/= ros (fix ros))
                        (setq ros (1+ (fix ros)))
                )
                        (setq sp1 (/ xss ros))
                        (setq i 1)
                    (while (<= i ros)
                        (setq ps9 (polar ps5 0 his))
                        (command "pline" ps5 ps9 "")
                        (command "_pedit" (polar ps5 0 0.025) "w"
0.0025"")
                        (setq ps5 (polar ps5 (/ pi -2) sp1))
                        (setq i (1+ i))
                    )
                        (command "pline" ps5 (polar ps5 0 his)"")
                        (command "_pedit" (polar ps5 0 (/ his 2)) "w"
0.0025"")

                    (command "zoom" "p")

                    (setq px8 (polar px1 (/ pi -2) 0.30))
                (setq px9 (polar (polar px8 0 (- x1 (/ wc 2))) (/ pi -2) (/ (- LS
wcz) 2)))
                    (setq pt15 (polar (polar (polar px2 (/ pi -2) 0.30) (/ pi -2)
LS) (/ pi -6) 0.80))
                    (command "pline" px8 (polar px2 (/ pi -2) 0.30) (polar
(polar px2 (/ pi -2) 0.30) (/ pi -2) LS)
```

```
                                        (polar px8 (/ pi -2) LS) px8"")
                        (command "pline" px9 (polar px9 0 wc) (polar (polar px9
0 wc) (/ pi -2) wcz)
                                        (polar px9 (/ pi -2) wcz) px9"")
                        (command "_pedit"  (polar px9 0 (/ wc 2)) "w" 0.007"")
  (command "text" (polar (polar px9 (/ pi -2) (* (/ wcz 3) 2)) pi (+ (- x1 (/ wc 2))
0.125)) 0.035 90 (rtos LS 2 2))
  (command "text" (polar (polar px9 0 (/ wc 3)) (/ pi 2) 0.025) 0.035 0.00 (rtos wc
2 2))
  (command "text" (polar (polar px9 (/ pi -2) (* (/ wcz 3) 2))  pi 0.025) 0.035 90
(rtos wcz 2 2))

    (if (/= y2 0.00)
      (command "pline"  px8 px9""
                "pline" (polar px2 (/ pi -2) 0.30) (polar px9 0 wc)""
                "pline" (polar (polar px2 (/ pi -2) 0.30) (/ pi -2) LS) (polar
(polar px9 0 wc) (/ pi -2) wcz)""
                "pline" (polar px8 (/ pi -2) LS) (polar px9 (/ pi -2) wcz)"")
    )
                        (command "zoom" "v")
  (command "pline" (polar px8 pi 0.075) (polar px8 pi 0.125) (polar px8 pi 0.10)
(polar (polar px8 pi 0.10) (/ pi 2) 0.025)
                        (polar (polar px8 pi 0.10 )(/ pi -2) (+ LS 0.025)) (polar
(polar px8 pi 0.10 )(/ pi -2) LS )
                (polar (polar (polar px8 pi 0.10 )(/ pi -2) LS ) pi 0.025)
                        (polar (polar (polar px8 pi 0.10 )(/ pi -2) LS ) 0 0.025)"")

    (initget 2 " 1 2 3 ")
            (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3 :"))
            (setq sc (atoi sca))
     (if (> sc 2)
              (command "text" pd11 0.04 0.00 "1:25")
          (if (< sc 2)
```

```
                    (command "text" pd11 0.04 0.00 "1:10")
                    (command "text" pd11 0.04 0.00 "1:20")
                )
            )
                (cond
                    (( = sca "1") (command "scale" "w" pt13 pt15 "" bm 10))
                    (( = sca "2") (command "scale" "w" pt13 pt15 "" bm 5))
                    (( = sca "3") (command "scale" "w" pt13 pt15 "" bm 4))
                )

                    (command "zoom" "A")
                    (princ)
                    (princ)
        ) ;end footings
    (setvar "blipmode" 1)
    (setvar "cmdecho" 1)
```

SHAPES.LSP

```
(vmon)
(defun c:a1 (/ pt1 pt2 ang )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nLeft point :"))
(setq pt2 (getpoint pt1 "\nRight point :"))
(setq ang ( angle pt1 pt2))
(command "pline" (polar pt1 (+ ang (/ pi 4)) (* 1.414 0.03)) (polar pt1 (+ ang (/
pi 2)) 0.03) "a" pt1 "l"
                        pt2 "a" (polar pt2 (+ ang (/ pi 2)) 0.03) "l" (polar pt2 (+
ang (- pi (/ pi 4)))(* 1.414 0.03))"")
(princ)
) ; end a1


(vmon)
(defun c:asc (/ pt1 pt2 ang )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nPoint 1:"))
(setq pt2 (getpoint pt1 "\nPoint 2:"))
(setq ang ( angle pt1 pt2))
(setq sca (getint "\nScale 1:10=10 ,1:20=5 ,1:25=4 ,1:100=<Enter> :"))
(if (not sca)(setq sca 1))
    (command "pline" (polar pt1 (+ ang (/ pi 4)) (* 1.414 (* sca 0.03))) (polar pt1
(+ ang (/ pi 2)) (* sca 0.03)) "a" pt1 "l"
                        pt2 "a" (polar pt2 (+ ang (/ pi 2))(* sca 0.03)) "l" (polar
pt2 (+ ang (- pi (/ pi 4)))(* 1.414 (* sca 0.03)))"")
(princ)
) ; end asc
```

```
(vmon)
(defun c:a2 (/ pt1 pt2 pt3 ang1 ang2 )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nFrist point :"))
(setq pt2 (getpoint pt1 "\nSecond point :"))
(setq ang2 (- (angle pt1 pt2) (/ pi 2)))
(command "pline" pt2 pt1 "a" (polar pt1 ang2 0.03) "l" (polar pt1 (+ ang2 (/ pi
4))(* 1.414 0.03))"")
(setq pt3 (getpoint pt2 "\nThird point :"))
(setq ang1 ( angle pt2 pt3 ))
(command "pline" pt2 pt3 "a" (polar pt3 (+ ang1 (/ pi -2)) 0.03) "l"  (polar pt3 (+
ang1 (/ pi 4) pi)(* 1.414 0.03))"")
(princ)
) ; end a2


(vmon)
(defun c:aa2 (/ pt1 pt2 pt3 ang1 ang2 )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
```

```
(setq pt1 (getpoint "\nFrist point :"))
(setq pt2 (getpoint pt1 "\nSecond point :"))
(setq ang1 (angle pt1 pt2))
(command "pline"  (polar pt1 (+ ang1 (/ pi 4)) (* 1.414 0.03)) (polar pt1 (+ ang1 (/
pi 2)) 0.03) "a" pt1"l" pt2"")
(setq pt3 (getpoint pt2 "\nThird point :"))
(command "pline" pt2 pt3"")
(setq pt4 (getpoint pt3 "\nEnd point :"))
(setq ang2 (angle pt3 pt4))
(command "pline" pt3 pt4 "a" (polar pt4 (+ ang2 (/ pi 2)) 0.03) (polar pt4 (+ ang2
(- pi (/ pi 4)))(* 1.414 0.03))"")
(princ)
) ; end aa2


(vmon)
(defun c:aaa2 (/ pt1 pt2 pt3 ang1 ang2 )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nFrist point <7 point> :"))
(setq pt2 (getpoint pt1 "\n2 point :"))
(setq ang1 (angle pt1 pt2))
(command "pline"  (polar pt1 (+ ang1 (/ pi 4)) (* 1.414 0.03)) (polar pt1 (+ ang1 (/
pi 2)) 0.03) "a" pt1"l" pt2"")
(setq pt3 (getpoint pt2 "\n3 point :"))
(command "pline" pt2 pt3"")
(setq pt4 (getpoint pt3 "\n4 point :"))
(command "pline" pt3 pt4"")
(setq pt5 (getpoint pt4 "\n5 point :"))
(command "pline" pt4 pt5"")
(setq pt6 (getpoint pt5 "\n6 point :"))
(command "pline" pt5 pt6"")
(setq pt7 (getpoint pt6 "\nEnd point :"))
```

```
(setq ang2 (angle pt6 pt7))
(command "pline" pt6 pt7 "a" (polar pt7 (+ ang2 (/ pi 2)) 0.03) (polar pt7 (+ ang2
(- pi (/ pi 4)))(* 1.414 0.03))"")
(princ)
) ; end aaa2



(vmon)
(defun c:a3 (/ pt1 pt2 ang )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nLeft point :"))
(setq pt2 (getpoint pt1 "\nRight point :"))
(setq ang ( angle pt1 pt2))
(command "pline" (polar pt1 (+ ang (/ pi -4)) (* 1.414 0.03))  (polar pt1 (+ ang (/
pi -2)) 0.03) "a" pt1 "l" pt2
                                    "a" (polar pt2 (+ ang (/ pi 2)) 0.03) (polar pt2 (+ ang
(- pi (/ pi 4)))(* 1.414 0.03))"")
(princ)
) ; end a3

(vmon)
(defun c:a4 (/ pt1 pt2 pt3 ang1 ang2 )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nFrist point :"))
(setq pt2 (getpoint pt1 "\nSecond point :"))
(setq ang2 (- (angle pt1 pt2) (/ pi 2)))
(command "pline" pt2 pt1 "a" (polar pt1 (+ pi ang2) 0.03) "l" (polar pt1 (+ ang2 (-
pi (/ pi 4)))(* 1.414 0.03))"")
(setq pt3 (getpoint pt2 "\nThird point :"))
(setq ang1 (angle pt2 pt3 ))
```

```
(command "pline" pt2 pt3 "a" (polar pt3 (+ ang1 (/ pi 2)) 0.03) "l" (polar pt3 (+
ang1 (- pi (/ pi 4))) (* 1.414 0.03))"")
(princ)
) ; end a4

(vmon)
(defun c:aa4 (/ pt1 pt2 pt3 ang1 ang2 )
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
(setq pt1 (getpoint "\nFrist point :"))
(setq pt2 (getpoint pt1 "\nSecond point :"))
(setq ang2 (- (angle pt1 pt2) (/ pi 2)))
(command "pline" pt2 pt1 "a" (polar pt1 ang2 0.03) "l" (polar pt1 (+ ang2 (/ pi
4))(* 1.414 0.03))"")
(setq pt3 (getpoint pt2 "\nThird point :"))
(setq ang1 ( angle pt2 pt3 ))
(command "pline" pt2 pt3 "a" (polar pt3 (+ ang1 (/ pi 2)) 0.03) "l" (polar pt3 (+
ang1 (- pi (/ pi 4))) (* 1.414 0.03))"")
(princ)
) ; end aa4


(vmon)
(defun c:a5 (/ pt1 pt2 pt3 spa hi l1 l2)
    (setvar "blipmode" 0)
    (setvar "cmdecho" 0)
        (setq pt1 (getpoint "\nFirst point :"))
        (setq pt3 (getpoint  pt1 "\nSecond point :"))
        (setq spa (getreal "\nSpacing stirrup :"))
        (setq hi (getreal "\nHight stirrup :"))
        (setq ang (angle pt1 pt3))
        (setq l1 (distance pt1 pt3))
```

```
        (setq l2 (- (car pt3)(car pt1)))
        (setq num (/ l2 spa))
1.   (if (/= num (fix num))
        (setq num (1+ (fix num)))
     )
        (setq spa (/ l1 num))
      (setq i 1)
     (while (<= i num)
        (setq pt2 (polar pt1 (/ pi 2) hi))
(command "pline" pt1 pt2 "")
        (setq pt1 (polar pt1 ang spa))
       (setq i (1+ i))
     )
(command "pline" pt1 (list (car pt1)(+ hi (cadr pt1))) "")
(princ)
) ; end a5

(vmon)
(defun c:a6 (/ pt1 hi ris tr nos num i pt2 pt3)
    (setvar "blipmode" 0)
    (setvar "cmdecho" 0)
        (setq pt1 (getpoint "\nBase point :"))
        (setq hi (getreal "\nFloor to floor height :"))
        (setq ris (getreal "\nRiser <0.175> :"))
        (setq tr (getreal "\nTread <0.25> :"))
        (setq nos (getreal "\nNoscing <0.025> :"))
        (if (not ris)
        (setq ris 0.175))
        (if (not tr)
        (setq tr 0.25))
        (if (not nos)
        (setq nos 0.025))
        (setq num (/ hi ris))
```

```
(if (/= num (fix num))
        (setq num (1+ (fix num)))
)
        (setq ris (/ hi num))
      (setq i 2)
    (while (<= i num)
        (setq pt2 (list (- (car pt1) nos)(+ (cadr pt1) ris)))
        (setq pt3 (list (+ (car pt1) tr nos)(+ ris (cadr pt1))))
(command "pline" pt1 pt2 pt3 "")
        (setq pt1 pt3)
      (setq i (1+ i))
    )
(command "pline" pt1 (list (- (car pt1) nos)(+ ris (cadr pt1))) "")
(princ)
) ;end a6
```

```
        (vmon)
        (defun c:a8 (/ bm p1 sp dia ra ang x1 ro i)
                (setvar "blipmode" 0)
                (setvar "cmdecho" 0)
                (setq bm (getpoint "\nFrist point :"))
                (setq p1 (getpoint bm "\nSecond point :"))
                (setq sp (getreal "\nSpacing steel <0.20> :"))
                (setq dia (getreal "\nDiameter <0.012> :"))
            (if (not dia)
                    (setq dia 0.012)
            )
                (setq ra (/ dia 2))
              (if (not sp)
                  (setq sp 0.20)
              )
```

```
            (setq ang (angle bm p1))
            (setq x1 (- (car p1)(car bm)))
        (if (= ang (/ pi 2))
            (setq x1 (- (cadr p1)(cadr bm)))
            (setq x1 (/ x1 (cos ang)))
        )
            (setq ro (/ x1 sp))
         (if (/= ro (fix ro))
            (setq ro (1+ (fix ro)))
         )
            (setq sp (/ x1 ro))
            (setq i 1)
        (while (<= i ro)
            (command "circle" bm ra)
            (setq bm (polar bm ang sp))
            (setq i (1+ i))
        )
            (command "circle" bm ra)
            (princ)
) ;end a8


        (vmon)
(defun c:aa8 ()
            (setvar "blipmode" 0)
            (setvar "cmdecho" 0)
            (setq bm (getpoint "\nFrist point :"))
            (setq p1 (getpoint bm "\nSecond point :"))
            (setq ns (getreal "\nNumber of steel  :"))
            (setq dia (getreal "\nDiameter <0.012> :"))
      (if (not dia)
            (setq dia 0.012)
      )
```

```
            (setq ra (/ dia 2))
            (setq ang (angle bm p1))
            (setq x1 (- (car p1)(car bm)))
          (if (= ang (/ pi 2))
            (setq x1 (- (cadr p1)(cadr bm)))
            (setq x1 (/ x1 (cos ang)))
          )
            (setq ro (- ns 1))
           (if (/= ro (fix ro))
            (setq ro (1+ (fix ro)))
           )
            (setq sp (/ x1 ro))
            (setq i 1)
          (while (<= i ro)
            (command "circle" bm ra)
            (setq bm (polar bm ang sp))
            (setq i (1+ i))
          )
            (command "circle" bm ra)
            (princ)
) ;end aa8


    (vmon)
    (defun c:a9 (/ bm p1 hi sp x1 ro ang i p2)
                    (setvar "blipmode" 0)
                    (setvar "cmdecho" 0)
                    (setq bm (getpoint "\nFrist point :"))
                    (setq p1 (getpoint bm "\nSecond point :"))
                    (setq hi (getdist p1 "\nHigh <0.50> :"))
                    (setq sp (getreal "\nSpacing steel <0.20> :"))
          (if (not hi)
```

```
                    (setq hi 0.50)
                )
                (if (not sp)
                    (setq sp 0.20)
                )
                    (setq ang (angle bm p1))
                    (setq x1 (- (car p1)(car bm)))
                  (if (= ang (/ pi 2))
                    (setq x1 (- (cadr p1)(cadr bm)))
                    (setq x1 (/ x1 (cos ang)))
                  )
                    (setq ro (/ x1 sp))
                (if (/= ro (fix ro))
                    (setq ro (1+ (fix ro)))
                )
                    (setq sp (/ x1 ro))
                    (setq i 1)
                (while (<= i ro)
                    (setq p2 (polar bm (+ (/ pi 2) ang) hi))
                    (command "pline" bm p2 "")
                    (setq bm (polar bm ang sp))
                    (setq i (1+ i))
                )
                    (command "pline" bm (polar bm (+ (/ pi 2) ang)
hi)"")
                    (princ)
            ) ;end a9

            (vmon)
(defun c:a10 (/ pt1 pt2 ang hk hk1 se sp ra dia dia1 pt3 pt4 x1 i dra)
            (setvar "blipmode" 0)
            (setvar "cmdecho" 0)
            (setq pt1 (getpoint "\nLeft point :"))
```

```
(setq pt2 (getpoint pt1 "\nRight point :"))
(setq ang (angle pt1 pt2))
(command "pline" pt1 pt2 "")

(if (not hk) (setq hk 0.03))
(setq hk1 hk)
(prompt "Lenght of hook <")
(princ hk)
(setq hk  (getreal "> :"))
(if (not hk) (setq hk hk1))

    (initget 2 "Y N")
      (setq se (getkword "\nDraw cross steel  (y/n):"))
    (cond
      ((= se "Y")
                    (setq sp (getreal "\nSpacing steel <0.20>
:"))
                    (if (not dia) (setq dia 0.012))
                    (setq dia1 dia)
                    (prompt "Diameter m<")
                    (princ dia)
                    (setq dia  (getreal "> :"))
                    (if (not dia) (setq dia dia1))

                (setq ra (/ dia 2))
                 (if (not sp)
                (setq sp 0.20)
                     )
                        (setq pt3 (polar pt1 (+ ang (/ pi 8)) (* 4 ra)))
                        (setq pt4 (polar pt2 (+ ang (- pi (/ pi 8))) (*
4 ra)))
                 (setq x1 (- (car pt4)(car pt3)))
                (if (= ang (/ pi 2))
```

```
                (setq x1 (- (cadr pt4)(cadr pt3)))
                (setq x1 (/ x1 (cos ang)))
                    )
                (setq ro (/ x1 sp))
                  (if (/= ro (fix ro))
                (setq ro (1+ (fix ro)))
                  )
                (setq sp (/ x1 ro))
                (setq i 1)
              (while (<= i ro)
                    (command "circle" pt3 ra)
                            (setq pt3 (polar pt3 ang sp))
                            (setq i (1+ i))
                      )
                            (command "circle" pt3 ra)
             )
                ((= se "N")
                            (setq dia 0.025)
                 )
           )

(initget 2 "A B C")
   (setq dra (getkword "\nAngle 45=A,90=B,135=C,<180=Enter>:"))
(cond
   ((= dra "A") (command "pline" pt1 (polar pt1 (+ ang (- pi (/ pi 4))) hk)""
                                "pline" pt2 (polar pt2 (+ ang (/ pi 4)) hk)"")
   )
   ((= dra "B") (command "pline" pt1 (polar pt1 (+ ang (/ pi 2)) hk)""
                                "pline" pt2 (polar pt2 (+ ang (/ pi 2)) hk)"")
   )
   ((= dra "C") (command "pline" pt1 (polar pt1 (+ ang (/ pi 4)) hk)""
                                "pline" pt2 (polar pt2 (+ ang (- pi (/ pi 4))) hk)"")
   )
```

```
    ((null dra)   (command "pline" pt2 "a" (polar pt2 (+ ang (/ pi 2)) (* dia 2.5))
(polar pt2 (+ ang (- pi (/ pi 4)))(* 1.414 (* dia 2.5)))""
                  "pline" pt1 "a" (polar pt1 (+ ang (/ pi 2)) (* dia 2.5)) (polar pt1
(+ ang (/ pi 4)) (* 1.414 (* dia 2.5)))"")
     )
   )
(princ)
) ;end a10


(vmon)
(defun c:a11 ()
(setq pt1 (getpoint  "\nStart point:"))
(setq pt2 (getpoint pt1  "\nEnter width:"))
(setq pt3 (getpoint pt2  "\nEnter height:"))
(setq w (distance pt1 pt2))
(setq h (distance pt2 pt3))
(setq a (* w h)) (terpri)
(princ a)
(princ)
) ; end a11

(defun c:a12 ()
    (setq ent (car (entsel "\nSelect entity:")))
    (entdel ent)
) ;end a12



(vmon)
(defun c:a13 ()
```

```
(setvar "blipmode" 0)
(setvar "cmdecho" 0)
  (initget 2 "2 3 4")
     (setq tye (getkword "\nType of steel 2,3,4 <1=Enter>:"))
  (cond
     ((= tye "2")
                (setq pt1 (getpoint "\nFrist point :"))
                (setq pt2 (getpoint pt1 "\nSecond point :"))
                (setq ang2 (- (angle pt1 pt2) (/ pi 2)))
                (command "pline" pt2 pt1 "a" (polar pt1 ang2 0.03) "l" (polar pt1
(+ ang2 (/ pi 4))(* 1.414 0.03))"")
                (setq pt3 (getpoint pt2 "\nThird point :"))
                (setq ang1 ( angle pt2 pt3 ))
                (command "pline" pt2 pt3 "a" (polar pt3 (+ ang1 (/ pi -2)) 0.03) "l"
(polar pt3 (+ ang1 (/ pi 4) pi)(* 1.414 0.03))"")
      )
      ((= tye "3")
                (setq pt1 (getpoint "\nFrist point :"))
                (setq pt2 (getpoint pt1 "\nSecond point :"))
                (setq ang1 (angle pt1 pt2))
                (command "pline"  (polar pt1 (+ ang1 (/ pi 4)) (* 1.414 0.03))
(polar pt1 (+ ang1 (/ pi 2)) 0.03) "a" pt1"l" pt2"")
                (setq pt3 (getpoint pt2 "\nThird point :"))
                (command "pline" pt2 pt3"")
                (setq pt4 (getpoint pt3 "\nEnd point :"))
                (setq ang2 (angle pt3 pt4))
                (command "pline" pt3 pt4 "a" (polar pt4 (+ ang2 (/ pi 2)) 0.03)
(polar pt4 (+ ang2 (- pi (/ pi 4)))(* 1.414 0.03))"")
      )

      ((= tye "4")
                (setq pt1 (getpoint "\nFrist point <7 point> :"))
                (setq pt2 (getpoint pt1 "\n2 point :"))
```

```
                (setq ang1 (angle pt1 pt2))
                (command "pline"  (polar pt1 (+ ang1 (/ pi 4)) (* 1.414 0.03))
(polar pt1 (+ ang1 (/ pi 2)) 0.03) "a" pt1"l" pt2"")
                (setq pt3 (getpoint pt2 "\n3 point :"))
                (command "pline" pt2 pt3"")
                (setq pt4 (getpoint pt3 "\n4 point :"))
                (command "pline" pt3 pt4"")
                (setq pt5 (getpoint pt4 "\n5 point :"))
                (command "pline" pt4 pt5"")
                (setq pt6 (getpoint pt5 "\n6 point :"))
                (command "pline" pt5 pt6"")
                (setq pt7 (getpoint pt6 "\nEnd point :"))
                (setq ang2 (angle pt6 pt7))
                (command "pline" pt6 pt7 "a" (polar pt7 (+ ang2 (/ pi 2)) 0.03)
(polar pt7 (+ ang2 (- pi (/ pi 4)))(* 1.414 0.03))"")
        )
                ((null tye)
                (setq pt1 (getpoint "\nLeft point :"))
                (setq pt2 (getpoint pt1 "\nRight point :"))
                (setq ang ( angle pt1 pt2))
                (command "pline" (polar pt1 (+ ang (/ pi 4)) (* 1.414 0.03))  (polar
pt1 (+ ang (/ pi 2)) 0.03) "a" pt1 "l"
                                        pt2 "a" (polar pt2 (+ ang (/ pi 2)) 0.03) "l"
(polar pt2 (+ ang (- pi (/ pi 4)))(* 1.414 0.03))"")
                )

    ) ; cond
      (princ)
 ) ;end a13


            (defun c:br (/ pt1 pt2 pt3 pt4  ang1 dst1)
                (setvar "osmode" 512)
```

```
1.          (setq pt1 (getpoint "\nSelect object :"))
            (setq pt2 (getpoint pt1 "\nEnter second point :"))
            (setvar "osmode" 128)
            (setq pt3 (getpoint pt1 "\nSelect parallel line :"))
            (setvar "osmode" 0)
            (setq ang1 (angle pt1 pt3))
            (setq dst1 (distance pt1 pt3))
            (setq pt4 (polar pt2 ang1 dst1))
                  (command "break" pt1 pt2
                  "break" pt3 pt4
                  "line" pt1 pt3""
                  "line" pt2 pt4""
               )
                        (princ)
      ) ; Break wall


(defun c:se (/ pt1 currnt last)
                  (setq pt1 (getpoint "\nPick start point :"))
            (setq spc (getdist pt1 "\nEnter number spacing :"))
            (setq currnt (getint "\nEnter first number :"))
            (setq last (getint "\nEnter last number :"))
            (setq stspc (rtos spc 2 2))
            (setq stspc (strcat "@" stspc "<0" ))
            (command "text" pt1 "" "" currnt)
               (repeat (- last currnt)
                  (setq currnt (1+ currnt))
                  (command "text" stspc "" "" currnt)
                    )
                     (princ)
) ; number of grid line
```

```
(defun c:imp (/ sp dt txt lns ls)
        (setq nme (getstring "\nName of text file to import :"))
        (setq sp (getpoint "\nText starting point :"))
        (setq lns (getdist "\nEnter line spacing in drawing units :"))
        (setq ls (rtos lns 2 2))
        (setq txt (open nme "r"))
        (setq dt (read-line txt))
        (setq ls (strcat "@" ls "<-90"))
        (command "text" sp "" "" dt)
       (while (/= dt nil)
        (setq dt (read-line txt))
        (command "text" ls "" "" dt)
       )
    (close txt)
    (command "redraw")

) ; Read data from file


        (vmon)
(defun c:sc ()
    (setq pt1 (getpoint "\nSelect point :"))
    (setq pt2 (getcorner pt1 "\nSelect corner :"))
        (command "zoom"  "w" pt1 pt2)
    (setq ang1 (angle pt1 pt2))
        (initget 2 " 1 2 3 ")
        (setq sca (getkword "\nScale 1:10=1,1:20=2,1:25=3 <no>:"))
       (cond
        (( = sca "1") (command "scale" "w"  pt1 pt2 "" pt1 10))
        (( = sca "2") (command "scale" "w"  pt1 pt2 "" pt1 5))
        (( = sca "3") (command "scale" "w"  pt1 pt2 "" pt1 4))
        (( null sca))
       )
```

```
(princ)
) ; end sc


        (vmon)
        (defun c:fin1 ()
                (setvar "blipmode" 0)
                (setvar "cmdecho" 0)
            (setq bm (getpoint "\nBase point lower beam :"))
            (setq hf1 (getdist bm "\nDepth of fin <0.60> :"))
                (if (not hf1)(setq hf1 0.60))
            (setq tf1 (getdist bm "\nThickness of fin <0.10>:"))
            (if (not tf1)(setq tf1 0.10))
            (setq co (getdist bm "\nCovering of concrete <0.025>:"))
            (if (not co)(setq co 0.025))
            (setq wb (getreal  "\nWidth of beam <0.20>:"))
            (if (not wb)(setq wb 0.20))
            (setq db (getreal  "\nDepth of beam <0.40> :"))
            (if (not db)(setq db 0.40))
            (setq pt1 (polar bm (/ pi -2) hf1))
            (setq pt2 (polar bm 0 wb))
                (setq pt3 (polar bm (/ pi 2) db))

            (command "pline" pt1 (polar pt1 0 tf1) (polar bm 0 tf1) pt2 (polar
pt2 (/ pi 2) db) pt3 pt1"")

                (command "pline" (polar (polar (polar pt1 (/ pi 2) (+ co
0.015)) 0 co)(/ pi 4)(* 1.414 0.03))
                        (polar (polar (polar pt1 (/ pi 2) (+ co 0.015)) 0 co) 0
0.03) "a"
                        (polar (polar pt1 (/ pi 2) (+ co 0.015)) 0 co) "l"
                        (polar (polar (polar pt1 (/ pi 2) (+ co 0.015)) 0 co) (/
pi 2) (- (+ hf1 db) (* co 3))) "a"
```

```
        (polar (polar (polar (polar pt1 (/ pi 2) (+ co 0.015)) 0 co) (/ pi 2) (- (+
hf1 db) (* co 3))) 0 0.03) "l"
        (polar (polar (polar (polar (polar pt1 (/ pi 2) (+ co 0.015)) 0 co)(/ pi 2) (-
(+ hf1 db) (* co 3))) 0 0.03)(/ pi -2) 0.03)"")
            (princ)
        )

        (vmon)
        (defun c:w1 (/ bm wi dep tw pt1 pt2 pt3 pt4 pt5)
          (setvar "cmdecho" 0)
          (setq rox 10)
          (setq xi 1)
          (while (<= xi rox)
           (if (= xi 1)
              (setq bm (getpoint "\nBase point :"))
               )
              (setq wi (getdist "\nWidth of column <0.40>:"))
               (if (not wi)(setq wi 0.40))
              (setq dep (getdist "\nDepth of column <0.80>:"))
            (if (not dep)(setq dep 0.80))
              (setq tw (getreal "\nThickness of wall <0.10>:"))
            (if (not tw)(setq tw 0.10))
              (setq pt6 (getpoint bm "\nTo point direction of column :"))
              (setq ang (angle bm pt6))
                 (setq len (distance bm pt6))
                 (setq pt1 (polar bm ang (/ wi 2)))
                 (setq pt2 (polar pt1 (+ ang (/ pi 2)) (/ dep 2)))
              (setq pt3 (polar pt2 (+ ang pi) wi))
              (setq pt4 (polar pt3 (+ (/ pi -2) ang) dep))
              (setq pt5 (polar pt4 ang wi))
                 (command "pline" pt2 pt3 pt4 pt5 pt2"")
```

```
            (setq bm (polar bm ang len))
                (setq xi (1+ xi))

        )

(princ)
) ; end w1
```