DEVELOPMENT OF A TEXT-BASED MATH BRAILLE TRANSLATION SOFTWARE TO ENHANCE LEARNING IN MATHEMATICS

WIRAMAN NIYOMPHOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (SCIENCE AND TECHNOLOGY EDUCATION) FACULTY OF GRADUATE STUDIES MAHIDOL UNIVERSITY 2010

Thesis
entitled

DEVELOPMENT OF A TEXT-BASED MATH BRAILLE TRANSLATION SOFTWARE TO ENHANCE LEARNING IN MATHEMATICS

……………….…………..…..………
Mr. Wiraman Niyomphol
Candidate

………………………..…..…..………
Assoc. Prof. Pintip Ruenwongsa, Ph.D.
Major advisor

………………………..…..…..………
Assoc. Prof. Bhinyo Panijpan, Ph.D.
Co-advisor

………………………..…..…..………
Miss. Patcharin Panjaburee, Ph.D.
Co-advisor

………………………..…..…..………
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

………………………..…..…..………
Assoc. Prof. Pintip Ruenwongsa, Ph.D.
Program Director
Doctor of Philosophy Program in
Science and Technology Education
Institute for Innovative Learning,
Mahidol University

Thesis

entitled

DEVELOPMENT OF A TEXT-BASED MATH BRAILLE TRANSLATION SOFTWARE TO ENHANCE LEARNING IN MATHEMATICS

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Doctor of Philosophy (Science and Technology Education)
on
August 16, 2010

…………….……….…..………
Mr. Wiraman Niyomphol
Candidate

……………….…..………………
Mr. Pisarn Soydhurum, Ph.D.
Chair

………………….…..…………..……
Miss. Patcharin Panjaburee, Ph.D.
Member

………………….…..………….……
Assoc. Prof. Pintip Ruenwongsa, Ph.D.
Member

………………….…..……………...…
Assoc. Prof. Bhinyo Panijpan, Ph.D.
Member

………………….…..………………..
Assoc. Prof. Anuchat Poungsomlee, Ph.D.
Acting Director
Institute for Innovative Learning
Mahidol University

………………….…..……………...…
Prof. Banchong Mahaisavariya, M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

………………….…..……………...…
Prof. Skorn Mongkolsuk, Ph.D.
Dean
Faculty of Science
Mahidol University

ACKNOWLEDGEMENTS

I would like to express my deep appreciation to my major advisor, Assoc. Prof. Pintip Ruenwongsa, Ph.D. for her guidance and supervision throughout the completion of this thesis. I am thankful for valuable comments and encouragement of my co-advisor, Assoc. Prof. Bhinyo Panijpan, Ph.D. as well as his wisdom and guidance on philosophy of Science. In addition, I am also grateful for my co-advisor, Lect. Patcharin Panjaburee, Ph.D. for her valuable comments and suggestions. I would like to extend special thanks to Mr. Pisarn Soydhurum, Ph.D. my chairperson of examination committee, for devoting his precious time and providing valuable comments.

I would like to thank Dr. Preedeeporn Limcharoen and Dr. Tusatrin Kruatong for their help and guidance during the earlier stage of my Ph.D. study. I also would like to extend my thanks for the invaluable friendship and support from my classmates and all staff at Institute for Innovative Learning with special thanks to my classmate and colleague Asst. Prof. Benjaporn Saksiri. Their guidance and assistance have made this thesis possible.

My utmost gratitude goes to Ratchasuda College and Ratchasuda Foundation and all staff for giving me the opportunity to enhance my career, always giving me support and encouragement.

Finally, I would like to thank my wife and my son for their love, understanding and encouragement as I was pursuing this important and challenging goal while carrying on with regular work and family matters.

Wiraman Niyomphol

DEVELOPMENT OF A TEXT-BASED MATH BRAILLE TRANSLATION SOFTWARE TO ENHANCE LEARNING IN MATHEMATICS

WIRAMAN NIYOMPHOL 4738681 ILSE/D

Ph.D. (SCIENCE AND TECHNOLOGY EDUCATION)

THESIS ADVISORY COMMITTEE: PINTIP RUENWONGSA, Ph.D., BHINYO PANIJPAN, Ph.D., PATCHARIN PANJABUREE, Ph.D.

ABSTRACT

This research project involved the development of a text-based math Braille translation software to support learning in mathematics. The researcher collected 192 math symbols from Nemeth Braille code resources. Nemeth code was used as the basis of text-based math symbols design for regular print text entry in any text editor. The math translation software was developed to convert these text-based math symbols to Braille. This software can be interfaced to Braille software for other languages. The software was tested for its performance with 10 subjects who are math teachers and material production staff in Thailand. They were given 3 hours of training on the use of the software and 2 hours to work on an exercise. Overall performance of the exercise was with 95.32 percent accuracy. Based on the questionnaire after the exercise, overall satisfaction level of using this method was in “acceptable” range. A follow-up interview reflects the opinion that this method would be most appropriate for math teachers and material production staff who are not familiar with Braille. This software should provide greater access to math education for children with visual impairment.

KEY WORDS: BRAILLE/ MATHEMATICS/ TRANSLATION/ NEMETH/ BLIND

140 pages

การพัฒนาโปรแกรมแปลอักษรเบรลล์คณิตศาสตร์แบบตัวพิมพ์เพื่อยกระดับการเรียนรู้ทางคณิตศาสตร์
DEVELOPMENT OF A TEXT-BASED MATH BRAILLE TRANSLATION SOFTWARE TO ENHANCE LEARNING IN MATHEMATICS

วีระแมน นิยมพล 4738681 ILSE/D

ปร.ด. (วิทยาศาสตร์และเทคโนโลยีศึกษา)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : พิณทิพ รื่นวงษา,Ph.D., ภิญโญ พานิชพันธ์,Ph.D., พัชรินทร์ ปัญจบุรี, Ph.D.

บทคัดย่อ

งานวิจัยนี้ได้พัฒนาโปรแกรมแปลอักษรเบรลล์แบบตัวพิมพ์เพื่อสนับสนุนการเรียนทางคณิตศาสตร์ ผู้วิจัยได้รวบรวมสัญลักษณ์คณิตศาสตร์จำนวน 192 สัญลักษณ์จากแหล่งข้อมูลรหัสเบรลล์ในระบบ Nemeth โดยระบบนี้ถูกนำมาใช้เป็นพื้นฐานของการออกแบบตัวพิมพ์สัญลักษณ์คณิตศาสตร์สำหรับตัวอักษรปกติซึ่งสามารถใช้กับโปรแกรมพิมพ์งานทั่วไปได้ โปรแกรมแปลอักษรเบรลล์คณิตศาสตร์ถูกพัฒนาขึ้นมาเพื่อเปลี่ยนสัญลักษณ์คณิตศาสตร์แบบตัวพิมพ์มาเป็นอักษรเบรลล์ โปรแกรมที่พัฒนาขึ้นมานี้สามารถนำมาเชื่อมกับโปรแกรมแปลอักษรเบรลล์สำหรับภาษาอื่นๆ โปรแกรมนี้ถูกนำมาทดสอบกับผู้ใช้ 10 คน ซึ่งเป็นครูคณิตศาสตร์และเจ้าหน้าที่ผลิตสื่อในประเทศไทย ผู้เข้าร่วมงานวิจัยในครั้งนี้ได้เข้าร่วมอบรมการใช้โปรแกรม จำนวน 3 ชั่วโมงและทำแบบฝึกหัดจำนวน 2 ชั่วโมง ผลจากการทำแบบฝึกหัดพบว่ามีความถูกต้อง 95.32 % ผลจากแบบสอบถามพบว่าผู้ใช้มีความพึงพอใจกับการพิมพ์สัญลักษณ์คณิตศาสตร์ด้วยวิธีดังกล่าว ในระดับ “ยอมรับได้” นอกจากนั้นความคิดเห็นของผู้ใช้จากการสัมภาษณ์เพิ่มเติมพบว่าวิธีนี้เหมาะสมมากที่สุดสำหรับครูคณิตศาสตร์และเจ้าหน้าที่ผลิตสื่อที่ไม่คุ้นกับการใช้อักษรเบรลล์ โปรแกรมนี้น่าจะช่วยให้นักเรียนพิการทางสายตาได้รับโอกาสการศึกษาทางคณิตศาสตร์มากยิ่งขึ้น

140 หน้า

CONTENTS

	Page
ACKNOWLEDGEMENTS	**iii**
ABSTRACT (ENGLISH)	**iv**
ABSTRACT (THAI)	**v**
LIST OF TABLES	**viii**
LIST OF FIGURES	**ix**
CHAPTER I INTRODUCTION	**1**
1.1 Background of the study	1
1.2 Objectives of the study	4
1.3 Research Questions	4
1.4 Research Instrument	4
1.5 Definitions	5
1.6 Expected Results	5
CHAPTER II LITERATURE REVIEW	**6**
2.1 Braille	6
2.2 English and Thai Braille	9
2.3 Braille for mathematics	15
2.4 Computer Braille	17
2.5 Software related to Braille	21
2.6 Math education of the blind	24
2.7 The National Center for Blind Youth in Science (NCBYS)	25
2.8 Math and science education in Thailand	26
2.9 Educational support	26
CHAPTER III METHODOLOGY	**30**
3.1 Software Development	30
3.2 Complete list of math symbols with text-based design	31
3.3 Development of the search table	51

CONTENTS (cont.)

		Page
3.4	Development of the source code	71
3.5	Performance Test	80
3.6	Attitude Test	81
CHAPTER IV	**RESULTS**	**82**
4.1	Software Development	82
4.2	Performance Test	92
4.3	Analysis of 20 math expressions N1 – N20	95
4.4	Analysis of 3 math questions	103
4.5	Attitude Test	106
CHAPTER V	**DISCUSSION**	**109**
5.1	Implications	112
5.2	Limitations	113
5.3	Recommendations for further studies	114
CHAPTER VI	**CONCLUSION**	**115**
6.1	Software development	116
6.2	Performance test and User opinion	117
6.3	Conclusion	118
REFERENCES		**120**
APPENDIX		**124**
BIOGRAPHY		**140**

LIST OF TABLES

Table		Page
3.1	Sample of Math Table with Print Typing Code and Braille Representations	31
4.1	Math Table with Print typing code and Braille representations (32 items for the exercise)	83
4.2	Twenty math expressions (before translation)	86
4.3	Three math questions consisting of text and expressions (before translation)	87
4.4	Translated result shown in US computer Braille and Braille font	89
4.5	Exercise Results of 20 math Expressions and 3 Math Questions	94
4.6	Summary of Exercise Results	95

LIST OF FIGURES

Figure		Page
2.1	illustrates a full cell of Braille	7
2.2	illustrates dots 1-3-4	7
2.3	illustrates the list of English alphabet a – z	9
2.4	illustrates the word “Thai” with capital “T”	10
2.5	illustrates The date “August 12, 2010”	10
2.6	illustrates an example of a Thai word “gin” meaning “to eat”	11
2.7	illustrates the complete list of English characters with numbers and punctuation marks	12
2.8	illustrates the complete list of Thai characters with numbers and punctuation marks	13
2.9	illustrates the comparison between High Numbers and Low Numbers	16
2.10	illustrates the expression one half plus x in print and Nemeth Braille	17
2.11	illustrates Computer keyboard layout with US Computer Braille in six dot format	20
2.12	illustrates appropriate dot pattern in Braille	20
2.13	illustrates the interface of Thai Braille Translation TBTW program	21

CHAPTER I
INTRODUCTION

1.1 Background of the study

An important part of quality learning environment for the blind that most people take for grant it is accessible books and learning materials. In light of providing equal educational opportunity for all children in the world, this has been a challenge for blind students for a long time. With advances in technology today, electronic books, audio books and Braille books can be produced more quickly. However, in mathematics subjects where blind students prefer to have books in Braille more than other formats, it is still difficult to find math Braille transcribers to produce math books fast enough to meet the demand. The use of technology for mathematics instruction enhances mathematical thinking, student and teacher discourse, and higher-order thinking by providing tools for exploration and discovery (Bitter and Hatfield, 1998). Many software developers contributed solutions to this problem by offering math Braille translation software. This research provides one more option to math Braille translation. This is purely a text-based solution where all math symbols are represented by using printable characters from the computer keyboard. The result of translation is math Braille code of NEMETH standard.

In reviewing different solutions to translate math symbols, there are many techniques ranging from text-based to graphics-based. Duxbury Braille Translator began with extra option in the menu bar where math functions were presented by names and desired values were inserted when each function was selected (Sullivan, 1997). Then additional solutions were created via third party tools like graphical math editor of Scientific Notebook and file importation of text-based format of LaTex (Sullivan, 2001). Recently, Duxbury offered the option to handle math symbols from Mathtype which is graphical tool available in Microsoft Word (Sullivan, 2008). WinBraille also has math translation capability built in the software. Mathematic symbols are inserted with Equation Editor 3.0 (Windows tool) in Word or WinBraille which is then translated to LaTex, NEMETH or Swedish Braille (Lofstedt, 2006).

WIMAT (Webel -ICEVI Mathematics Transcription Software) is another graphical math translation software with its own editor. It was a collaborative project between ICEVI and Webel software development company introduced at ICEVI World Conference in 2006. English uncontracted Braille is supported. The user can select math function from the menu, insert proper value and appropriate Nemeth Braille will be produced on the editor along with other text (Mani, 2006).

Text-based print representation of math symbols has been around in many platforms. Basic symbols are already available on the computer keyboard such as decimal point, comma, equals, less than, greater than, and different types of parenthesis. Operators such as plus, minus, multiplication and division are generally denoted by +, -, *, / respectively. LaTex and Megadots are examples of text-based approach.

In order to represent additional math functions, LaTex text publishing system originated from New Mexico University offers a very good text-based solution. One can write a math document consisting of text and math equations by using word description to represent function names such as SQRT for square root. Finally, the entire text of math document can be printed and actual graphical math symbols can be read by a sighted person. Even though LaTex does not concentrate on math Braille translation, both Duxbury and WinBraille did take LaTex into account and supported it further for Braille translation.

Another Braille translator that became popular during DOS days is Megadots. It is now Windows-based and provides math translation capabilities (Holiday, 1998). While the basic concept is text-based, Megadots attempts to deal with printable symbols from the keyboard more than using textual description like LaTex. In addition, math symbols can be selected from pull-down menus. For example, a simple fraction of one half would be written as [1//2]. One needs to have some understanding of math Braille code while typing print character representation. In print, only number 1 and 2 would be written with a line indicating the fraction. For Braille however, the idea of having opening and closing fraction indicators is necessary as well.

Each system reviewed above seems to have its usefulness in its own rights and may be popular or unused or outdated depending on the development in technology and may be used within certain group of people. There are both graphics-based and text-based solutions. There are both accessible and inaccessible solutions by visually impaired persons.

This research is offering another alternative to text-based solution. The objective is to develop a software module or engine where all math symbols can be typed via regular keyboard to translate into math code in Nemeth system. Nemeth is one Braille code standard for math and scientific notations which is used widely. This math software engine can be tied to software application which handles other issues such as Windows interface, file handling, editing, embossing, etc but translation portion, especially translation of math equations is left to the task of this engine.

In order to deal with all symbols, the researcher had to consider how to represent all math symbols with regular characters. The aim was to use as intuitive way of representation as possible by using two or three combination of punctuations and letters. For example, x^2”+1 would represent x squared follows by plus 1 where quotation mark is used to mark the end of exponential value or back to base line level. On the other hand x^n+1 would represent x to the power of n+1.

Although this software module can be plugged into other Braille translation software of other languages, it was implemented in the context of a Thai Braille translation software called RS Braille (Niyomphol and Tandayya and Nantachaipitak, 2008). RS Braille attempted to deal with MathML translation to Braille but did not have text-based math Braille translation capability until interfaced with the software from this research. Previous work for Thai language includes TBT (Thai Braille Translator) developed under DOS (Niyomphol, 1989). TBT translation engine was further developed and implemented under Windows interface as Thai Braille Translation Software for Windows (Niyomphol and Nantachaipitak, 2002). Both TBT and TBTW do not have math translation capability.

After the software was designed and developed, this text-based math Braille translation software was evaluated for its performance with math teachers and material production staff in Thailand by letting them work on exercises after a training session. The discussion of the results of questionnaires and interviews completed by those users were gathered and analyzed as part of this research.

1.2 Objectives of the study

1. To develop text-based math Braille translation software module and interface it with a Thai Braille translation software
2. To find out the performance of this software among math teachers and material production staff as well as their opinion in order to facilitate mathematics learning process of children with visual impairment.

1.3 Research Questions

1. Can math symbols according to Nemeth Braille code be represented with print characters and a text-based math Braille translation software be developed based on that design?
2. What is the performance of this text-based math Braille translation software when tested with math teachers and material production staff?
3. What is the attitude of math teachers and material production staff after using this text-based math Braille translation software?

1.4 Research Instrument

1. A Thai Braille translation software (RS Braille) properly interfaced with Text-based Math Braille translation engine developed under this research project.
2. Math exercises consisting of 20 math expressions and 3 math questions that include both text and math expressions.
3. Attitude test questionnaire.

1.5 Definitions

1. Braille – A reading/writing system for the blind by using combination of raised dots within a matrix shape of 3 rows and 2 columns.
2. Nemeth Braille code – A standard for writing Braille math and scientific notations developed by Abraham Nemeth.
3. Braille translation software – A computer program that converts print text of a particular language or a set of print symbols into Braille.
4. US Computer Braille – A standard of United States to represent all printable characters from computer keyboard used to display Braille information via a Braille printer or a refreshable Braille display.
5. Text-based math input method – A technique of representing print characters and math symbols by using only available printable characters from the computer keyboard.
6. Material production staff – Staff in the field of visual impairment responsible for producing learning material for visually impaired students in Braille, large-print, audio or tactile formats.

1.6 Expected Results

1. A purely text-based math Braille translation software was developed to handle input of text and math expressions in order to convert into math Braille code according to Nemeth standard. Given text-based nature of the software, coded math books to be produced in Braille can be typed via any basic text editor.
2. Results of the performance of this text-based math Braille translation software when tested with math teachers and material production staff along with their attitude regarding this method.
3. Application of the software to enhance learning of blind students by making Braille math materials more available and increase learning interaction between blind students and math teachers.

CHAPTER II
LITERATURE REVIEW

In order to support the work done in this research project, Basic information about Braille, various Braille translation software with special focus on math translation, and challenges in learning mathematics of visually impaired students were gathered as follows.

2.1 Braille

Does anyone really know how many braille codes exist? Louis Braille would be amazed and probably very pleased that his creation provided the means for literacy for people who are blind all over the world. What would he think about the proliferation of codes that present barriers to communication among blind people in different countries or even in different regions of the same country? Some of the changes made to the braille code have been necessary to allow braille to be adapted to languages other than its original French, but those altered codes retained the concept of using upper cells for letters and numbers and lower cells for auxiliary symbols, such as punctuation, in accordance with Louis Braille's original design (Bogart, 2009).

Braille system was invented by Louis Braille in 1824. Louis Braille was born on January 4, 1809 in the city of Coupvray in France. He died on January 6, 1852 at the age of 43. His invention became a worldwide system used by blind and visually impaired people for reading and writing. There are numerous sources of information about six dot structure and his life. Braille is read by passing the fingers over characters made up of an arrangement of one to six embossed points. It has been adapted to almost every known language in the world. The dots are arranged as a matrix of 3 rows of 2 columns. A full cell of Braille is shown in Figure 2.1.

Dot 1 – Top left	Dot 4 – Top right
Dot 2 – Middle left	Dot 5 – Middle right
Dot 3 – bottom left	Dot 6 – Bottom right

Figure 2.1 illustrates a full cell of Braille

Each dot is identified by a number 1 through 6. It is very important to refer to dots shown in a Braille cell by its proper number. Braille devised this system in 1821. Each Braille character or cell is made up of six dot positions, arranged in a rectangle containing two columns of three dots each. A dot may be raised at any of the six positions to form sixty-four (26) possible subsets, including the arrangement in which no dots are raised. For reference purposes, a particular dot pattern may be described by naming the positions where dots are raised, the positions being universally numbered 1 to 3, from top to bottom, on the left, and 4 to 6, from top to bottom, on the right. As shown in Figure 2.2, dots 1-3-4 would describe a cell with three dots raised, at the top and bottom in the left column and on top of the right column. This is letter “m”. The lines of horizontal Braille text are separated by a space, much like visible printed text, so that the dots of one line can be differentiated from the Braille text above and below. Punctuation is represented by its own unique set of characters. Louis Braille’s 6 dot system is accepted as the invention that revolutionized written communication for the blind.

Figure 2.2 illustrates dots 1-3-4

Louis Braille’s life According to National Braille Press can be summarized as follows:

Louis' father was a saddle maker. He became blind at the age of three, when he accidentally poked himself in the eye with a stitching awl, one of his father's workshop tools. At the age of 10, Braille earned a scholarship to the National Institute for the Blind in Paris, one of the first school for the blind in the world.

Originally, the students were taught how to read by feeling raised letters (a system devised by the school's founder, Valentin Hauy). However, because the raised letters were made using paper pressed against copper wire, the students never learned to write. Another disadvantage of this system was that the letters weighed a lot and whenever people published books using this system, they put together a book with multiple stories in one in order to save money. This made the books sometimes weigh over a hundred pounds. The school had just 14 books, all of which Louis had read.

In 1821, Charles Barbier, a former Captain in the French Army, visited the school. Barbier shared his invention called "night writing" a code of 12 raised dots and a number of dashes that let soldiers share top-secret information on the battlefield without having to speak. The code was too difficult for Louis to understand and he later changed the number of raised dots to 6 to form what we today call Braille.

In the same year, Louis Braille began inventing his raised-dot system with his father's stitching awl, the same instrument with which he had blinded himself, finishing at age 15, in 1824. Inspired by the wooden dice his father gave to him, his system used only six dots and corresponded to letters, whereas Barbier's used 12. The six-dot system allowed the recognition of letters with a single fingertip apprehending all the dots at once, requiring no movement or repositioning which slowed recognition in systems requiring more dots. These dots consisted of patterns in order to keep the system easy to learn. The Braille system also offered numerous benefits over Hauy's raised letter method, the most notable being the ability to both read and write an alphabet.

Braille later extended his system to include notation for mathematics and music. The first book in Braille was published in 1829 under the title Method of Writing Words, Music, and Plain Songs by Means of Dots, for Use by the Blind and Arranged for Them.

Braille became a well-respected teacher at the Institute. Although he was admired and respected by his pupils, his writing system was not taught at the Institute during his lifetime. He died in Paris of tuberculosis in 1852 at the age of 43. His system was finally officially recognized in France two years after his death, in 1854 (National Braille Press, 2009).

Today, blind students use different tools to write Braille. For low-cost portable device, a slate would be used. Another device that helps blind children write more quickly without using high-tech electronics equipment is to use a Braillewriter. This is equivalent to using a typewriter for a sighted person. For high-tech solutions, a blind person may use an electronic Braille note taker which allows data input via a Braille keyboard and Braille data can be stored in the device.

2.2 English and Thai Braille

Over the past two hundred years, Braille has been the preferred writing/reading method for the blind throughout the world. Many languages has adopted Braille system and developed Braille standard for characters in their languages. Braille system was adopted in France and the system was used by other countries throughout Europe including England. English Braille was adopted in 1932 by the United States (Grey, 2009). Figure 2.3 shows the list of English alphabet a – z.

a	b	c	d	e	f	g	h	i	J
⠁	⠃	⠉	⠙	⠑	⠋	⠛	⠓	⠊	⠚
k	l	m	n	o	p	q	r	s	t
⠅	⠇	⠍	⠝	⠕	⠏	⠟	⠗	⠎	⠞
u	v	w	x	y	z				
⠥	⠧	⠺	⠭	⠽	⠵				

Figure 2.3 illustrates the list of English alphabet a – z

By identifying dot location, it can be seen that “w” uses dots 2-4-5-6 and “n” uses dots 1-3-4-5. IN order to represent a capital letter, an extra cell of dot 6 is placed in front of the letter. The word “Thai” with capital “T” is shown in Figure 2.4.

	T	h	a	i
⠠	⠞	⠓	⠁	⠊

Figure 2.4 illustrates the word “Thai” with capital “T”

Numbers are similar to first 10 letters. Letters a, b, c, d, e, f, g, h, I, j are also used to represent numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 0. This is referred to as high numbers because all 10 digits use only the upper 4 dots of the matrix (dots 1-2-4-5). To distinguish between letters and numbers, a number sign is placed in front of a number. Number sign is represented by dots 3-4-5-6. In case of writing a numeric value of multiple digits, number sign is only used once in front of all digits. The date “August 12, 2010” is shown in Figure 2.5.

	A	u	g	u	s	t		1	2		2	0	1	0
⠠	⠁	⠥	⠛	⠥	⠎	⠞	⠼	⠁	⠃	⠼	⠃	⠚	⠁	⠚

Figure 2.5 illustrates The date “August 12, 2010”

After Braille system spread from Europe to United States, a graduate of Overbrook School for the Blind of Philadelphia Pennsylvania Genevieve Caulfield came to Thailand in 1938 to establish the first school for the blind (Inaram, 2004). Miss Caulfield worked with colleagues in Thailand to develop Thai Braille code where some Thai characters can still be traced to English alphabets of similar sounds these days (Niyomphol, 2009). The first Thai character (Gaw Gai) uses dots 1-2-4-5 which is the same as dot pattern for letter “g”. Thai character (Naw Noo) uses dots 1-3-4-5 which is the same as letter “n”. Figure 2.6 shows an example of a Thai word “gin” meaning “to eat”. The vowel used for this word (Sara ie) uses dots 1-2. Even though some Thai vowels can be placed above or below the main character of a word, Braille writing does not use multiple levels so all upper and lower vowels in Thai are placed along other characters in the same line.

กิ		น
⠛	⠃	⠝

Figure 2.6 illustrates an example of a Thai word “gin” meaning “to eat”

Figures 2.7 and 2.8 show the complete list of English and Thai characters with numbers and punctuation marks.

Since there are only 64 combination of dot pattern from sic dots (2 to the power of 6), it is necessary to use the same dot pattern for characters of other languages. Even within the same language, multiple cell technique would be used for some characters. English language would use two cells to indicate capital letters, Thai language would use two cells to indicate extra character of identical sound, and numbers would use number sign in front of the value of the number.

In order to reduce number of spaces used to write Braille, the idea of contractions or shorthand was introduced. English language has English Braille level 2 which uses many contractions. Thai language contracted some vowels in Thai Braille level 1 and developed Thai Braille level 2 which has many contractions as well.

While Braille in other languages continue to develop, English speaking countries began unification process of English Braille code (UEB). Based on the first report of the committee studying the possibility of unification in November 1992, presented at the Celebration of the Bicentenary of Louis Braille's birth at UNESCO Headquarter, Paris, presented by Joe Sullivan, some insights gained from that effort reflects back to the great contribution of Louis Braille.

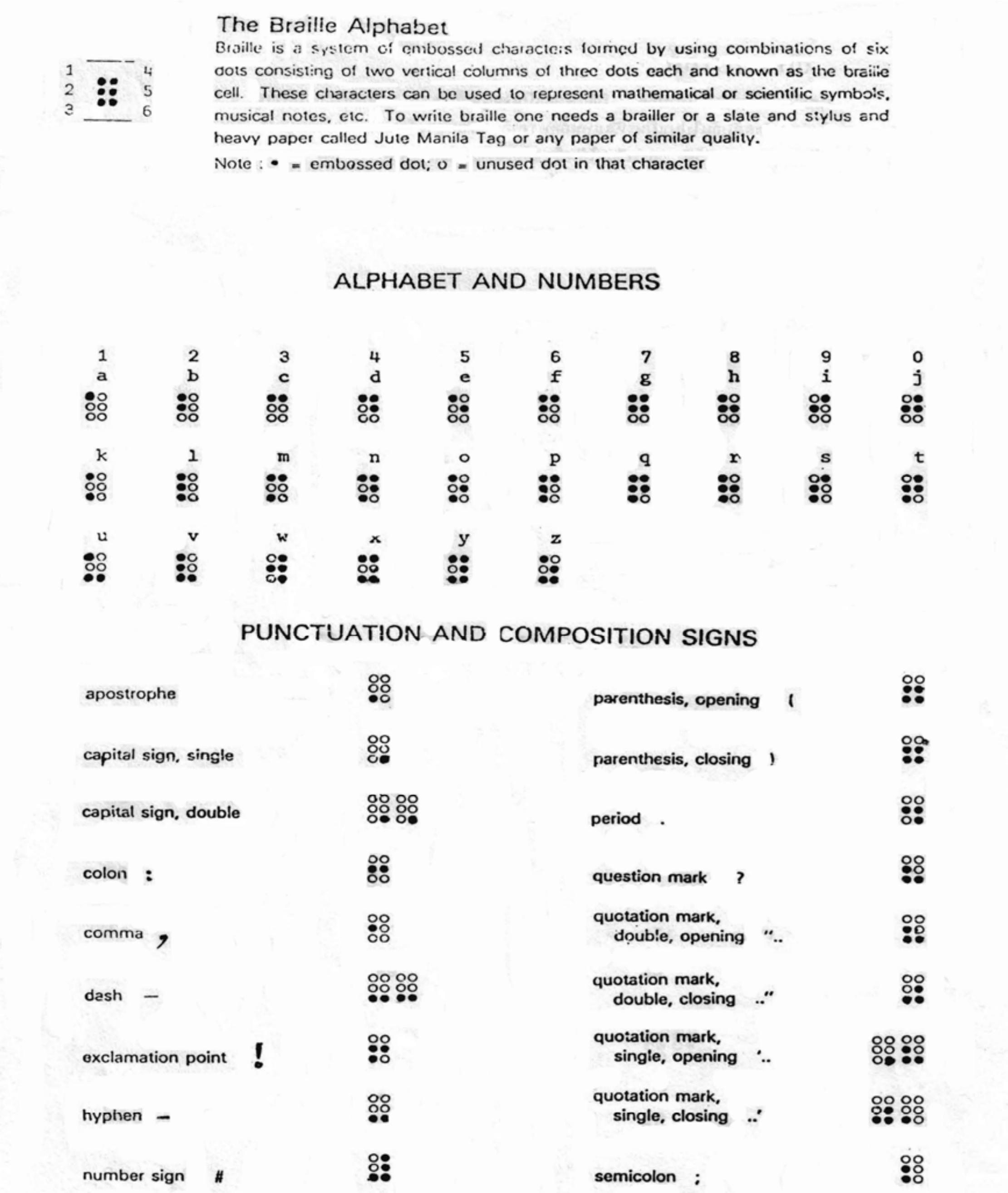


Figure 2.7 illustrates the complete list of English characters with numbers and punctuation marks

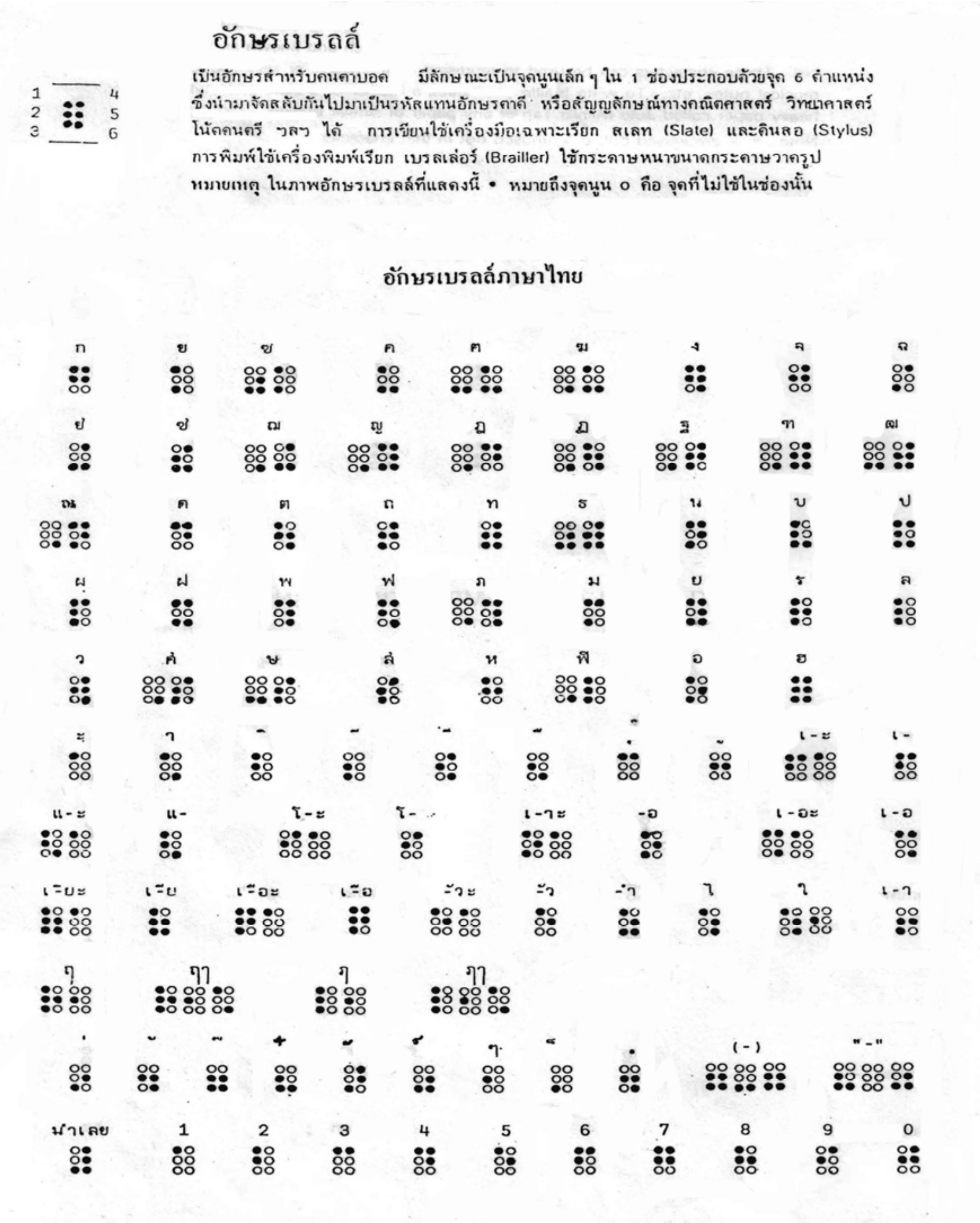


Figure 2.8 illustrates the complete list of Thai characters with numbers and punctuation marks

The committee realized that there is a fundamental need to ensure that the extent of any one braille symbol, whether it be a single-cell or a multi-cell symbol, can be readily determined, even if the meaning of the symbol is not immediately known. Otherwise, the reader who encounters an unfamiliar symbol cannot be sure what to look up and ambiguities are bound to arise when a certain sequence of single-cell signs could also be interpreted as a multi-cell sign. This need is realized in UEB by symbol formation rules based upon a "prefix-root" structure that flows naturally from the principles evident in Louis Braille's original design. This structure is one of the cornerstones of UEB that gives it extensibility as well as clarity.

In addition, the assignment of symbols for the digits, and the general rules surrounding numbers, must be decided early in the process because everything else is affected by those decisions.

Finally, the use of contractions in literary braille, which typically takes up all the single-cell symbols that are available after the alphabet, digits and basic punctuation are assigned, makes it hugely complicated to add capabilities for mathematics and science in a way that is both unambiguous and efficient.

The committee found that that unification is technically difficult, and that Louis Braille’s basic design principles and original decisions most often provided the best guidance to a way forward (Sullivan, 2009).

Given various issues involved in Braille code, one may think that it would be very confusing for the blind to read Braille. However, the blind learned to observe the context of information being read and could distinguish between different languages and notice when encountering numbers, punctuation marks as well as contractions. Over the years, Braille for other subjects were developed such as music notations, math symbols, scientific notations, computer Braille, and phonetics. Therefore, the sic dot system of Braille has survived for over two hundred years and is still the preferred choice of writing and reading for the blind today.

2.3 Braille for mathematics

For mathematics and scientific notations, Dr. Abraham Nemeth developed the system and it was called Nemeth Braille in recognition of his name. Although numbers and some basic symbols can be written using High Numbers As part of normal literary text, NEMETH Braille code is essential in writing math expressions. Math books for blind students from primary, secondary to college levels are produced using NEMETH. Science subjects like Physics and Chemistry also rely on NEMETH for expressing formulas.

According to Acharya (2009), Braille codes for Mathematics based on six dot system has been effectively used for representing the symbols used in Mathematics. This would help a visually impaired person learn mathematics using standard Braille. The assignment of codes for mathematical symbols was proposed in 1952 by Dr. Abraham Nemeth, himself blind from birth but rose to the position of a Professor Emeritus of Mathematics at the University of Detroit in the United States. Prof. Nemeth's system has gained wide acceptance in the world and the system of Braille Codes has been named "Nemeth Braille" in his honor. Approximately 190 assignments were classified into groups based on specific topics in mathematics representing:

- Numerals and Numbers
- signs of Operation (Symbols for Operators)
- Comparisons, Ratios and Proportions
- Symbols used in Set Theory
- Different types of Parentheses
- Mathematical Symbols and Constants
- Shapes and Relating to Shapes
- Polygons
- Greek Letters (Acharya, 2009)

In NEMETH system, numbers are written as Low Number. This technique moves High Numbers down one level and uses only bottom 4 dots (dots 2-3-5-6). Number 1 would be dot 2 instead of 1, and number 0 would be dots 3-5-6 instead of dots 2-4-5. IN this way, there would be no confusion between letters and numbers especially when using letters a – j. Expression “2b” would use dots 2-3 for number 2 and dots 1-2 for letter b. Figure 2.9 shows the comparison between High Numbers and Low Numbers without using number sign and an example of a number written with a letter.

	1	2	3	4	5	6	7	8	9	0
High Numbers	⠁	⠃	⠉	⠙	⠑	⠋	⠛	⠓	⠊	⠚
Low Numbers	⠂	⠆	⠒	⠲	⠢	⠖	⠶	⠦	⠔	⠴

	2	b	+	3	c
⠼	⠆	⠃	⠬	⠒	⠉

Figure 2.9 illustrates the comparison between High Numbers and Low Numbers

Since Braille is a linear writing/reading technique and Nemeth is based on Braille, Nemeth system is also a linear system. According to Dot Less Braille, the need to convey mathematics linearly, without the use of special typesetting and often with a limited character set, is a common one. Linear mathematics is necessary for computer codes from FORTRAN to Java, for HTML, for braille, and for numerous other applications. The Nemeth code solves the problem of linear mathematics in a particularly elegant and compact manner as compared to other linear math entry methods (Dot Less Braille, 2009).

Because Braille is a linear writing method, NEMETH developed extra Braille symbols to mark the beginning and ending of some math concepts to make sure there is no confusion when reading them. A fraction of one half would have open fraction sign, number 1, fraction line, number 2, and close fraction sign. With close fraction sign after number 2, expression one half plus x would be understood that plus operation must be applied to one half instead of having x become part of the denominator 2+x. Open fraction sign uses dots 1-4-5-6, fraction line uses dots 3-4, and close fraction sign uses dots 3-4-5-6. Figure 2.10 shows the expression one half plus x in print and Nemeth Braille.

	1	/	2		+	x
⠹	⠂	⠌	⠆	⠼	⠬	⠭

Figure 2.10 illustrates the expression one half plus x in print and Nemeth Braille

There are other math standards for Braille. RNIB math Braille code from Royal National Institute for the Blind in England is another system that is used in many countries. In contrary to sighted students where math symbols are international standard, blind students growing up learning math would not be able to understand math Braille code from books produced with different standard (Sullivan, 2009). Since Nemeth Braille code is widely used as standard math code in America and other countries including Thailand, this research is implemented according to NEMETH standard.

2.4 Computer Braille

In 1983, the Braille Authority of North America (BANA), under the chairmanship of Richard Evensen, took on the challenge of creating a braille code for computers. The committee that was formed to create Computer Braille Code (CBC) was Chaired by Tim Cranmer, a coauthor and distributor of public-domain braille translation software, and other notable members as Priscilla Harris, from the National Braille Association, and Joseph Sullivan, from Duxbury Systems (Grey, 2009).

With advances in computer technology especially in 1980's, the blind had the opportunity to learn how to use computer and different Braille devices were produced to work with computer. Braille printers or Braille embossers were produced to help produce file into hardcopy via a Braille printer. A blind user can listen to information typed from computer keyboard and screen messages by using a Screen Reading software. In addition, a blind user who is familiar with Braille can use a Braille display device which can be attached to the computer under the keyboard and refreshable electronic Braille can be displayed From Braille cells. To display Braille in print for sighted persons, many Braille fonts based on Computer Braille were developed. Therefore, various Braille computer peripherals and Braille software use Computer Braille to represent characters.

The use of Braille printers and Braille displays led to the development of Computer Braille standard and eight dot Braille. Two more dots would be used along with existing 6 dot matrix. A dot would be placed under dot 3 (bottom left) and is referred to as dot 7. Another dot would be placed under dot 6 (bottom right) and is referred to as dot 7. Eight dot Braille is more widely used among computer users that rely on the Braille display especially blind programmers. However, Braille printers are still being used to produce Braille hardcopy in six dot. Braille font also use six dot. Many countries especially in United States and Europe developed their own Computer Braille standard. Therefore, US or North America Computer Braille would be different from German Computer Braille. One may think that computer Braille must be utilized in eight dot format only. However, it is possible to write Computer Braille in six dot format to express technical information. Because of its unique set of dot pattern that can represent all 64 Braille combination differently, US Computer Braille became the standard Braille code for many programmers when working with Braille translation software, Braille file and sending data to a Braille embosser.

Regardless of using eight or six dot system, Braille displays, Braille printers and Braille fonts are still interfaced to the computer on the computer by Computer Braille standard which is different from English literary Braille and NEMETH Braille code. Furthermore, different languages especially European languages, developed their own Computer Braille standards. For this research, US Computer Braille or North America Computer Braille will be used.

The aim of Computer Braille is to have only one Braille pattern using one Braille cell representing one character on the computer keyboard. This is strictly one-to-one correspondence between print and Braille. All alphabets a – z, numbers 1 – 0, and all punctuation marks on the computer keyboard would be uniquely represented in Braille. Dots 7 and 8 are used to help in this concept. While small letters a – z use combination of six dots, capital letters A – Z would use the same dot pattern but add dot 7 below previous pattern. Small “a” uses dot 1 but capital “A” use dots 1-7. For US computer Braille, 1 – 0 are written as Low Numbers, just like NEMETH system without using number sign. Other symbols or punctuation marks on the keyboard use different dot patterns and there are no duplications at all. Figure 2.11 shows Computer keyboard layout with US Computer Braille in six dot format

1	2	3	4	5	6	7	8	9	0	-	=	
⠂	⠆	⠒	⠲	⠢	⠖	⠶	⠦	⠔	⠴	⠤	⠿	
!	@	#	$	%	^	&	*	(	)	_	+	
⠮	⠈	⠼	⠫	⠩	⠘	⠯	⠡	⠷	⠾	⠸	⠬	
q	w	e	r	t	y	u	i	o	p	[	]	\
⠟	⠺	⠑	⠗	⠞	⠽	⠥	⠊	⠕	⠏	⠪	⠻	⠳
Q	W	E	R	T	Y	U	I	O	P	{	}	|
⠟	⠺	⠑	⠗	⠞	⠽	⠥	⠊	⠕	⠏	⠪	⠻	⠳
a	s	d	f	g	h	j	k	l	;	‘		
⠁	⠎	⠙	⠋	⠛	⠓	⠚	⠅	⠇	⠰	□		
A	S	D	F	G	H	J	K	L	:	“		
⠁	⠎	⠙	⠋	⠛	⠓	⠚	⠅	⠇	⠱	□		

z	x	c	v	b	n	m	,	.	/			
⠵	⠭	⠉	⠧	⠃	⠝	⠍	⠠	⠨	⠌			
Z	X	C	V	B	N	M	<	>	?			
⠵	⠭	⠉	⠧	⠃	⠝	⠍	⠣	⠜	⠹			

Figure 2.11 illustrates Computer keyboard layout with US Computer Braille in six dot format

By using Computer Braille, a desired dot pattern can be sent to the Braille display, Braille printer or shown on the screen via Braille font by sending or typing the appropriate character. Since dot 6 is comma in Computer Braille, to represent capital T (dots 6, dots 2-3-4-5), one can type ",t" in US Computer Braille to display capital "T" in literary English Braille. One half in NEMETH can be typed as "?1/2#" because question mark would be dots 1-4-5-6 (open fraction sign) and number sign would be dots 3-4-5-6 (close fraction sign). If "?1/2#" is changed to Braille font, one would see appropriate dot pattern in Braille as shown in Figure 2.12

?	1	/	2	#
⠹	⠂	⠌	⠆	⠼

Figure 2.12 illustrates appropriate dot pattern in Braille

Therefore, Computer Braille has become the preferred choice to represent Braille information as the background code when working with computer equipment or software application. The end result in Braille whether it is Braille in a particular language or math Braille symbols, is left to the process of Braille translation software.

2.5 Software related to Braille

Braille translation software from print to Braille has been around for a long time. As the first commercial English-based Braille translation software, Duxbury Braille Translator (DBT) from Duxbury Systems Inc. was developed in 1975. Some other translators are WinBraille from Index Braille, Megadots from Duxbury Systems Inc. and NFB Trans from National Federation of the Blind. Besides focusing on just translation of text information, other subject areas have been considered as well. Mathematics, Chemistry, Music, Phonetics, and Braille in various languages continued to bring challenges for Braille translation development. For Thai language, TBT (Thai Braille Translator) was first developed under DOS (Niyomphol, 1989). TBT translation engine was further developed and implemented under Window interface as TBTW (Niyomphol and Nantachaipitak, 2002). It is true that competent Braille transcribers can type everything in Braille directly via 6-key entry method. However, it is generally agreed that translation software helps speed the process of Braille production. For complicated content, one with knowledge of Braille for that particular area can work on the translated Braille file to produce more accurate results. Figure 2.13 is the snap shot of Thai Braille Translation TBTW program.

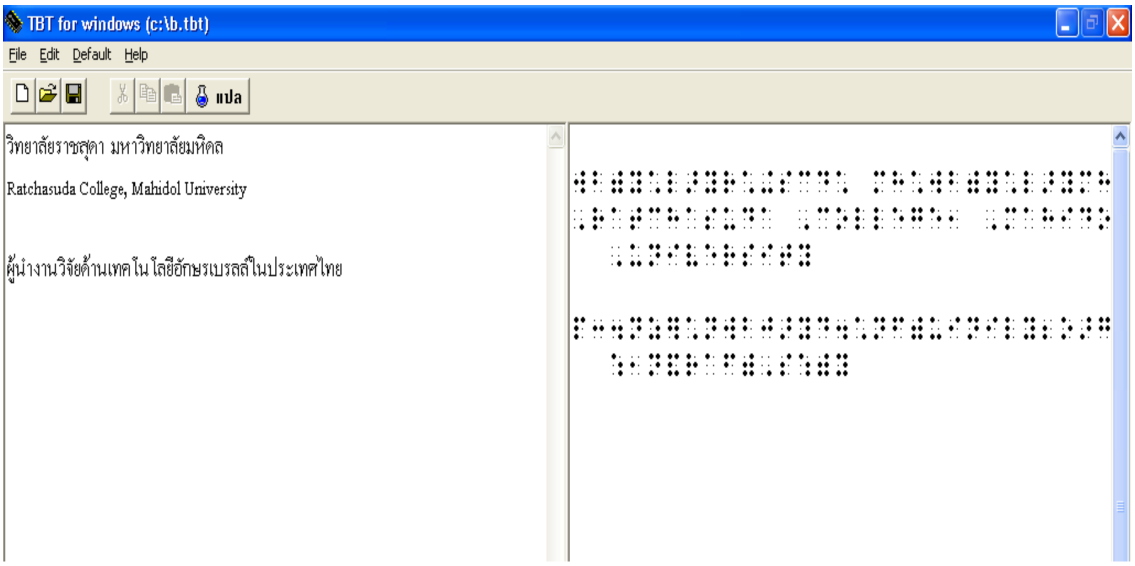


Figure 2.13 illustrates the interface of Thai Braille Translation TBTW program

For mathematics, there are many techniques ranging from text-based to graphics-based. Duxbury Braille Translator began with extra option in the menu bar where math functions are presented by names and desired values are inserted when the function is selected (Sullivan, 1997). Then additional solutions were created via third party tools like graphical math editor of Scientific Notebook and file importation of text-based format of LaTex (Sullivan, 2001). Recently, Duxbury offers way to handle math symbols from Mathtype which is graphical tool available in Microsoft Word (Sullivan, 2008). WinBraille also has math translation capability built in the software. The mathematics are inserted with Equation Editor 3.0 (Windows tool) in Word or WinBraille which is then translated to LaTex, Nemeth or Swedish Braille (Lofstedt, 2006).

WIMAT (Webel - ICEVI Mathematics Transcription Software) is another graphical math translation software with it's own editor. It was a collaborative project between ICEVI and Webel software development company introduced at ICEVI World Conference in 2006. English uncontracted Braille is supported. The user can select math function from the menu, insert proper value and appropriate Nemeth Braille will be produced on the editor along with other text (Mani, 2006).

Text-based math Braille translation or print representation has been around in many platforms. Basic symbols are already available on the keyboard such as decimal point, comma, equals, less than, greater than, and different types of parenthesis. Operators such as plus, minus, multiply and divide are generally denoted by +, -, *, / respectively. However, using just one character from the keyboard to represent math is limited.

In order to represent additional math functions, LaTex text publishing system originated from New Mexico University offers a very good text-based solution. One can write a math document consisting of text and math equation by using word description to represent function name such as sqrt for square root. Finally, the entire text and math document can be printed and show actual graphical math symbols for sighted person to read. Even though LaTex does not concentrate on math Braille translation, both Duxbury and WinBraille did take LaTex into account and supported it further for Braille translation.

Another Braille translator that became popular during DOS days is Megadots. It is now Windows based and provides math translation capabilities (Holiday, 1998). The basic idea is text-based but Megadots attempts to deal with printable symbols from the keyboard more than using textual description like LaTex. In addition, math symbols can be selected from pull-down menus. For example, a simple fraction of one half would be written as [1//2]. One needs to have some understanding of math braille code while typing print character representation. In print, only number 1 and 2 would be written with a line indicating the fraction. For Braille However, the idea of having open fraction and closing fraction indicators are necessary as well.

Equation editor or Mathtype in Microsoft Word has become a popular tool for sighted person to produce document with math content. Both Duxbury and WinBraille were improved to support translation of math formulas from Microsoft to Braille. Another Braille production solution that has gained popularity recently is Tiger system from ViewPlus Technologies. The Braille printer has Braille capability, tactile graphics, and combining both print and Braille on the same hardcopy output. The software for this system is TSS (Tiger Software Suite). TSS has developed support for math content via Microsoft Word as well.

Among all graphics solution that support math Braille translation, it is not possible for the visually impaired to use the software independently. Furthermore, not all math equations produced by these graphics tools can convert the content to Braille completely. Some equations may not be translated and questions marks will be given as the result. This means Braille production staff cannot rely on graphics tools alone. They must review the result on the Braille editor and make further editing/corrections as needed.

Each system seems to have its usefulness in its own rights and may be popular or unused or outdated depending on the development in technology and may be used within certain group of people. There are both graphics-based and text-based solutions. There are both accessible and inaccessible solutions by visually impaired persons.

Without taking credits away from all the work of graphics tools, text-based solution an alternative which is accessible by the visually impaired and can convert from print to Braille quite accurately. Megadots is a commercial solution for text-based math Braille translation which requires selection of math symbols from the menu. However, this research project is purely a text-based solution where regular text with math content can be typed in the same document by using any basic text editor.

2.6 Math education of the blind

Braille is an important part of education of the blind. According to a recent report by the National Federation of the Blind, less than 10 percent of the 1.3 million legally blind people in the United States can read Braille. Additionally, the report says only about 10 percent of blind children are learning it (National Federation of the Blind, 2009). According to Marty McKenzie, vision consultant, braille readers statistically have a much higher chance of gaining employment than others in the blind community. However, only 10 percent of blind students read Braille. Therefore, it is important to start learning early while they are in school (McKenzie, 2009). Mathematics is part of curriculum for students from elementary through High School and beyond. Braille math books are essential part of learning for blind students. Even though there are many ways to produce math Braille materials ranging from direct Braille transcription, Braille translation, or combination of both, blind students still face many challenges.

Although many blind individuals have succeeded in studying math and science and held careers in these fields, education is still a challenged for many blind children especially in non-developed countries. The software developed in this research project will contribute toward solving this problem by making math Braille books more available. This section presents a new initiative by National Federation for the Blind in United States (NFB) and a project in Thailand to promote blind students to study in math and Science track.

2.7 The National Center for Blind Youth in Science (NCBYS)

For too long blind youth have been denied full participation in public school science classrooms. Are dissections too dangerous? Should blind youth sit on the sidelines rather than participate and make their own observations? Is it too difficult to understand the stars if you cannot visually observe them in the sky? The answer to these questions is most certainly "no," but all too often parents, teachers, and blind students are simply not aware of how a blind person might actively participate in these activities. Insufficient hands-on materials, few teachers who understand tactile learning, and lack of access to blind adult role models and resources have placed artificial barriers on blind youth in the sciences. The NFB Jernigan Institute recognizes the urgent need to address these artificial barriers and equip regular education teachers, parents, and blind youth themselves with the tools and knowledge to provide greater opportunities in science to blind youth across the nation.

In order to dramatically change the opportunities and resources available to blind youth in science, technology, engineering, and math (STEM) subjects and careers, the National Federation of the Blind Jernigan Institute has initiated a bold national vision for the next generation of blind explorers - the National Center for Blind Youth in Science (NCBYS). The goals of the NCBYS are to:

- Establish a center of excellence for resources and information about how blind youth can best learn and understand scientific and mathematical concepts.
- Promote opportunities for blind youth in science through demonstration projects, partnerships, and public education.
- Provide technical support to projects working to improve materials and instruction for blind youth in science and math.
- Provide a national mentoring program built on the foundations established by the blind professionals of the NFB Science and Engineering Division.
- Centralize research previously completed in this area and coordinate ongoing research efforts.

The work of the center began in the summer of 2004 with the NFB Science Academy, the first ever series of summer programs developed and directed by the blind to teach science to blind youth. The Academy was a fantastic success, encouraging and inspiring all who participated. It was the most dynamic gathering of blind youth in the United States. Blind and low vision high school students from across the United States took part in this five-day academy to engage, inspire, and encourage the next generation of blind youth to consider careers falsely believed to be impossible for the blind (National Federation for the Blind, 2004).

2.8 Math and science education in Thailand

The challenge is not faced by only developed country like United States. Although many blind individuals in Thailand have graduated from universities, no one has graduated in STEM subject areas (Science, Technology, Engineering, Mathematics). Because of the lack of materials and proper tools along with the attitude from teachers that the blind should be discouraged from learning mathematics, blind children in Thailand lost the opportunity to develop themselves from very young age. Without a strong foundation, blind children all chose to study in Arts track instead of Science track when they arrive at grade 10.

A national initiative to promote math/science education for the blind began in 2006 with two blind students. Currently they are studying Computer Science in a University. With the support of Her Royal Highness Princess Sirindtorn and several leading organizations in Thailand, 9 blind students are now studying in math/science fields ranging from grade 10 to university (Niyomphol, 2010).

2.9 Educational support

Education for the blind has evolved from residential schools to integration in regular schools. Even though residential schools still exist to serve blind students with additional disabilities and act as resources for the larger integration system, developed countries concentrated more on integration or inclusion. Thailand has 13 schools for the blind, mostly act as educational center with living facility with integration to nearby public schools. Full integration where blind students live at home and attend schools near home are happening but students mostly do not receive proper support. Whatever the case may be, it is necessary to provide proper support to the student which involves vision teachers, regular classroom teachers, parents, and other partners. Following is a reference to the study on “Best Practices” for all parties by Carol Farrenkopf, Coordinator of the large Vision Program in Toronto, Ontario, Canada, which supports over 400 children who are visually impaired. Even though the information may not apply fully with Thai culture considering the communities and available resources, it is a good example for Thailand to select appropriate aspects and strive toward reaching higher goals in providing effective education services for the blind.

Everyone has a role to play in teaching children who are visually impaired to read. "Everyone" includes the classroom teacher, the teacher of students who are visually impaired, educational assistants or para-educators, certified orientation and mobility (O&M) specialists, and family members. Formal literacy instruction typically takes place during the school day, but parents and other family members may provide informal instruction within the home and community. The key is to expose children who are visually impaired to a rich variety of literacy experiences at every opportunity, whether in or out of school. Intensity of instruction and duration of instruction needs to be commensurate with students' needs. This Vision Program in Toronto, Ontario, Canada, supports over 400 children who are visually impaired. There are 18 qualified teachers of visually impaired students, 4 certified O&M specialists, a certified braille transcriber, and a braille librarian on staff. There are also several educational assistants who provide additional supports to students in the program (Farrenkopf, 2008).

In Ontario, "the first consideration regarding placement for an 'exceptional pupil' is placement in a regular class with appropriate supports, when such placement meets the student's needs and is in accordance with parents' wishes" (Ontario Ministry of Education, 2005).

Koenig and Holbrook (2000) and Corn and Koenig (2002) attempted to achieve a consensus among 40 experts in the field of visual impairment on how literacy instruction needs to be delivered to students who read braille and students with low vision. Four factors were considered in both studies: consistency of service (daily contact, moderate contact of 1 to 3 visits per week, low contact of semimonthly or monthly visits, or periodic contact of occasional visits throughout the year); total time per day (1 to 2 hours per session, a half-hour to 1 hour per session, or less than a half-hour per session); time span (infancy through grade 12); and duration of service (throughout one school year, throughout one quarter or semester, or concentrated in one to several intensive days of service).

The results indicated that the experts agreed that literacy instruction should vary in "consistency and intensity, depending on the skill area being addressed" (Koenig & Holbrook, 2000). For example, children who are just beginning to learn braille should have daily, intensive instruction for several years. Similarly, students with low vision should have "high-quality direct instruction in literacy skills" (Corn & Koenig, 2002) rather than mere consultative support.

Students who are blind or who have low vision deserve high-quality literacy instruction not only from their classroom teachers, but also from qualified teachers of visually impaired students and other significant individuals in their lives. When teaching a young child to read braille, the teacher of visually impaired students is not simply teaching a code-he or she is teaching reading at the same time. Literacy instruction cannot be separated from braille instruction; the two are intertwined. For the student with low vision who is just learning to read, learning how to integrate visual skills into literacy experiences is critical. Students with low vision also require direct instruction to learn how to use appropriate low vision devices (for example, monoculars or magnifiers) and high-tech devices (for example, closed-circuit televisions or computers with screen enlargement or speech output software) in order to access the school curriculum as independently as possible (D'Andrea & Farrenkopf, 2000).

Given the unique, disability-specific needs of learners who read braille or enlarged print, it is unrealistic to expect the regular classroom teacher to teach literacy skills without the assistance of a teacher of students with visual impairments. In addition, it is also unrealistic to expect that the teacher of students with visual impairments should be able to take on the entire literacy program for a young braille learner without the assistance of the regular classroom teacher. Families must also contribute to the process at home by reinforcing what is learned at school and by exposing their children to new and varied experiences within the community. We need to work together to develop the best possible literacy program for the student. Mutual respect of each other's skills is essential-none of us can provide a quality literacy program in isolation (Farrenkopf, 2008).

It can be seen that there are many factors that contribute toward the success of education of the blind. This text-based math Braille translation software interfaced to a Thai Braille translation software is a small but important part of the effort to provide support in the area of Braille production especially in math and science subjects.

CHAPTER III
METHODOLOGY

This research involved three steps. (1) Development of text-based math Braille translation software, (2) testing the performance of the software and the ability of the participants to analyze print math expressions during the exercise (3) find out the attitude of the participants using the software.

3.1 Software Development

In order to develop the software, a list of available math codes was gathered from Nemeth Reference and other resources (American Printing House for the Blind, 1972; Mani, 2005; Inaram, 2004). The total of 192 math symbols was found. Then the researcher designed print representation of these math symbols by using combination of printable characters on the keyboard. For example, one asterisk * is used for dot multiplication sign and two asterisks ** are used for cross multiplication sign. Table 3.1 is a sample of symbols from the complete list that were used during the test of the software. Notice that there are no print symbols for open fraction and close fraction (items 089 and 090) because such symbols are not used when writing fraction in print but they are necessary when writing in Braille. A Braille cell comprises of 6 dots arranged in 2 columns and 3 rows matrix where each dot has a number for identification. Dots 1, 2, 3 are at the left column from top to bottom. Similarly, dots 4, 5, 6 are at the right column from top to bottom. Plus sign uses dots 3-4-6. With this method, a sighted math teacher or material production staff without knowledge of Braille can type “Print typing code” and the software will convert to “Braille representation” properly.

Table 3.1 Sample of Math Table with Print Typing Code and Braille Representations

Item	Symbol	Description	Print Typing Code	Braille Representation
001	+	Plus Operator	+	
003	x	Multiplication cross ×	**	
004	•	Multiplication dot	*	
089		simple fraction indicator, opening	@(	
090		simple fraction indicator, closing	)@	
091	–	simple fraction line, horizontal	/	

3.2 Complete list of math symbols with text-based design

All 192 symbols were first identified by numbers 001 – 192where each symbol was given English and Thai description. Then the text-based representation of each symbol was designed to correspond to Braille code in US computer Braille format. Following is the list where each symbol is presented in five-line format.

Line 1: Item
Line 2: English Description
Line 3: Thai Description
Line 4: Print Typing Code
Line 5: Braille Representation

** Begin of list

001 plus operator **เครื่องหมายบวก** + +	006 division linear **เครื่องหมายหารยาว** -/ O
002 minus operator **เครื่องหมายลบ** - -	007 division spatial **เครื่องหมายหาร ว่าง** -/- 333
003 multiplication x **เครื่องหมายคูณ เอ็กซ์** ** @*	008 division slash **ทับ** / _/
004 multiplication dot **เครื่องหมายคูณ จุด** * *	009 plus or minus **เครื่องหมายบวกหรือลบ** +|- +-
005 division **เครื่องหมายหาร** // ./	010 minus or plus **เครื่องหมายลบหรือบวก** -|+ -+

011 plus followed by minus **เครื่องหมายบวกและลบ** +&- +"-	016 tilde **นิเสธ** ~ @:
012 minus followed by plus **เครื่องหมายลบและบวก** -&+ -"+	017 logical sum **โลจิคอล ผลรวม** @ls @+
013 minus followed by minus **เครื่องหมายลบและลบ** -&- -"-	018 logical product **โลจิคอล ผลคูณ** @lp @%
014 intersection **อินเตอร์เซ็กชั่น** *: .%	019 ampersand **เครื่องหมายแอนด์** & _&
015 union **ยูเนียน** *+ .+	020 vertical bar **โดยที่** | \

021 hollow dot **จุดกรวง** .o .*	026 less than with bar under **เครื่องหมายน้อยกว่าหรือเท่ากับ** <- "k:
022 backslash **เครื่องหมายแบคแสลท** \ _*	027 less than with equal sign under **เครื่องหมายน้อยกว่าหรือเท่ากับ (ด้านล่าง)** <= "k.k
023 equals **เครื่องหมายเท่ากับ** = .k	028 greater than with bar under **เครื่องหมายมากกว่าหรือเท่ากับ** >- .1:
024 less than **เครื่องหมายน้อยกว่า** < "k	029 greater than with equal sign under **เครื่องหมายมากกว่าหรือเท่ากับ (ด้านล่าง)** >= .1.k
025 greater than **เครื่องหมายมากกว่า** > .1	030 unequal **เครื่องหมายไม่เท่ากับ** /= /.k

<table>
<tr><td>031
less than followed by greater than
เครื่องหมายน้อยกว่าและมากกว่า
<>
"k".1</td><td>036
not perpendicular to
ไม่ตั้งฉาก
/|_
/$p</td></tr>
<tr><td>032
greater than followed by less than
เครื่องหมายมากกว่าและน้อยกว่า
><
.1""k</td><td>037
not parallel to
ไม่ขนาน
/|=
/$l</td></tr>
<tr><td>033
equal sign with question mark over
เครื่องหมายเท่ากับและเครื่องหมายคำถาม
?=
".k<_8]</td><td>038
ratio
อัตราส่วน
:
"1</td></tr>
<tr><td>034
perpendicular to
ตั้งฉาก
|_
$p</td><td>039
proportion
สัดส่วน
::
;2</td></tr>
<tr><td>035
parallel to
ขนาน
|=
$l</td><td>040
double tilde
โดยประมาณ
~~
@:@:</td></tr>
</table>

041 tilde with equal sign under เท่ากันทุกประการ ~= @:.k	046 equivalence จำนวนเท่ากัน == @<,<
042 arc, concave upward เส้นโค้งคว่ำ !^ $a	047 identity สมนัยกัน &id _|
043 are, concave downward เส้นโค้งหงาย !v $'	048 variation แปรผัน =* _=
044 is an element of เป็นสมาชิกของ &e @e	049 inclusion เป็นสับเซตของ =& _"k
045 reverse membership สมาชิกผกผัน /&e @5	050 reverse inclusion ไม่เป็นสับเซตของ /=& _.1

051 inclusion with bar under **เครื่องหมายประกอบด้วย** =&- _"k:	056 union with bar under **ยูเนียน มีเส้นด่านล่าง** *+- .+:
052 inclusion with equal sign under **เครื่องหมายประกอบด้วย, เท่ากับด้านล่าง** =&= _"k.k	057 contracted form of right-pointing arrow **ลูกศรชี้ขวา ขนาดเล็ก** -> $O
053 reverse inclusion with bar under **เครื่องหมายประกอบด้วย, เส้นด้านกลับ** =&/- _.1:	058 ordinary shaft, right-pointing arrow **ลูกศรชี้ขวา ปกติ** -h> $33O
054 reverse inclusion with equal sign under **เครื่องหมายเส้นด้านกลับ, เท่ากับด้านล่าง** =&/= _.1.k	059 double shaft, right-pointing arrow **ลูกศรชี้ขวาสองเท่า** -hh> $77O
055 intersection with bar under **อินเตอร์เซ็กชั่น มีเส้นด้านล่าง** *:- .%:	060 ordinary shaft, left-pointing arrow **ลูกศรชี้ซ้าย ปกติ** -h< $[33

<table>
<tr><td>061
double shaft, left-pointing arrow
ลูกศรชี้ซ้าย สองเท่า
-hh<
$[77</td><td>066
right-pointing over left-pointing arrow
ลูกศรชี้ขวา อยู่เหนือลูกศรชี้ซ้าย
-><
$33O$[33</td></tr>
<tr><td>062
arrow pointing up
ลูกศรชี้ขึ้น
|^
$<33O</td><td>067
left-pointing over right-pointing arrow
ลูกศรชี้ซ้าย อยู่เหนือลูกศรชี้ขวา
-<>
$[33$33O</td></tr>
<tr><td>063
arrow pointing down
ลูกศรชี้ลง
|v
$%33O</td><td>068
down-pointing followed by up-pointing arrow
ลูกศรชี้ลง ลูกศรชี้ขึ้น
-v^
$%33O"$<33O</td></tr>
<tr><td>064
ordinary shaft, double-pointing arrow
ลูกศรซ้ายขวา
-h<>
$[33O</td><td>069
up-pointing followed by down-pointing arrow
ลูกศรชี้ขึ้น ลูกศรชี้ลง
^v-
$<33O"$%33O</td></tr>
<tr><td>065
double shaft, double-pointing arrow
ลูกศรซ้ายขวาสองเท่า
-hh<>
$[77O</td><td>070
arrow pointing up and down
ลูกศรขึ้น และลง
-^v
$<[33O</td></tr>
</table>

071 parenthesis, left เครื่องหมายวงเล็บเปิด ((	076 brace, right วงเล็บปีกกาปิด } .)
072 parenthesis, right เครื่องหมายวงเล็บปิด))	077 angle bracket, left วงเล็บมุมเปิด &[..(
073 bracket, left วงเล็บก้ามปูเปิด [@(	078 angle bracket, right วงเล็บมุมปิด]& ..)
074 bracket, right วงเล็บก้ามปูปิด] @)	079 vertical bar, double เส้นตั้งตรงสองเส้น || \\
075 brace, left วงเล็บปีกกาเปิด { .(	080 enlarged parenthesis, left วงเล็บใหญ่เปิด *(,(

<table>
<tr><td>081
enlarged parenthesis, right
วงเล็บใหญ่ปิด
)*
,)</td><td>086
enlarged vertical bar, left and right
เส้นตรงหนาซ้ายและขวา
*|
,\</td></tr>
<tr><td>082
enlarged bracket, left
วงเล็บก้ามปูใหญ่เปิด
*[
@,(</td><td>087
empty set
เซ็ตว่าง
{}
.(.)</td></tr>
<tr><td>083
enlarged bracket, right
วงเล็บก้ามปูใหญ่ปิด
]*
@,)</td><td>088
transcriber's grouping symbol, opening & closing
สัญลักษณ์เปิดปิดกลุ่ม
"
,'</td></tr>
<tr><td>084
enlarged brace, left
วงเล็บปีกกาใหญ่เปิด
*{
.,(</td><td>089
simple fraction indicator, opening
เครื่องหมายเปิดเศษส่วน
@(
?</td></tr>
<tr><td>085
enlarged brace, right
วงเล็บปีกกาใหญ่ปิด
}*
.,)</td><td>090
simple fraction indicator, closing
เครื่องหมายปิดเศษส่วน
)@
#</td></tr>
</table>

091 simple fraction line, horizontal เส้นแสดงเศษส่วน / /	096 complex fraction indicator, closing ปิดเศษส่วนซ้อน }@ ,#
092 simple fraction line, diagonal เส้นแสดงเศษส่วนเอียง |/ _/	097 complex fraction line, horizontal เส้นแสดงเศษส่วนซ้อน @/ ,/
093 fractional part of mixed number indicator, opening เปิดเศษส่วนคระ @[_?	098 complex fraction line, diagonal เส้นแสดงเศษส่วนซ้อนเอียง @|/ ,_/
094 fractional part of mixed number indicator, closing ปิดเศษส่วนคระ]@ _#	099 shape indicator regular เครื่องหมายแสดงรูปทรง ปกติ ?r $
095 complex fraction indicator, opening เปิดเศษส่วนซ้อน @{,?	100 shaded shape indicator (bold) เครื่องหมายแสดงรูปทรง หนา ?b

101 filled-in shape indicator **เครื่องหมายแสดงรูปทรง ทึบ** ?f --	106 rectangle **รูปสี่เหลี่ยมผืนผ้า** !r $r
102 circle **รูปวงกลม** !c $c	107 diamond **รูปสี่เหลี่ยมขนมเปียกปูน** !d $d
103 triangle **รูปสามเหลี่ยม** !t $t	108 pentagon **รูปห้าเหลี่ยม** !p $5
104 right triangle **รูปสามเหลี่ยมมุมฉาก** !tr $t.r]	109 hexagon **รูปหกเหลี่ยม** !h $6
105 square **รูปสี่เหลี่ยมจัตุรัส** !s $4	110 star shape **รูปดาว** ?* $s

111 ellipse รูปวงรี !e $e	116 circle with interior dot จุดภายในวงกลม !c. $c_$*]
112 angle มุม ![$[	117 alpha แอลฟา 'a .a
113 right angle มุมฉาก ![r $[.r]	118 beta เบตา 'b .b
114 bar, horizontal, modifying เส้นแนวนอน !- :	119 gamma แกมมา 'g .g
115 circle with interior plus sign เครื่องหมายบวกภายในวงกลม !c+ $c_$+]	120 delta เดลตา 'd .d

121 epsilon เอปซิลอน 'e .e	126 kappa แคปปา 'k .k
122 zeta เซตตา 'z .z	127 lambda แลมดา 'l .l
123 eta เอตา 'et .:	128 mu มิว 'm .m
124 theta ธีตา 'th .?	129 nu นิว 'n .n
125 iota ไอโอตา 'i .i	130 xi ซาย 'x .x

131 omicron โอไมครอน 'o .o	136 upsilon อุปซิลอน 'u .u
132 pi พาย 'p .p	137 phi ฟาย 'ph .f
133 rho โร 'r .r	138 chi คาย 'ch .&
134 sigma ซิกมา 's .s	139 psi พซาย 'ps .y
135 tau ทาว 't .t	140 omega โอเมกา 'om .w

141 sampi **แซมพาย** 'c .c	146 subscript indicator **เครื่องหมายห้อย** ; ;
142 numeric indicator **เครื่องหมายนำเลข** ?# #	147 comma or comma-space within subscript or superscript **คอมม่าภายในเครื่องหมายยกกำลังหรือห้อย** _, [
143 letter sign **เครื่องหมายนำตัวอักษร** ?; ;	148 baseline indicator **เครื่องหมายนำแถวปกติ** " "
144 punctuation indicator **เครื่องหมายนำสัญลักษณ์** ?_ _	149 cancellation indicator, opening **เครื่องหมายเปิดยกเลิก** @# [
145 superscript indicator **เครื่องหมายยกกำลัง** ^ ^	150 cancellation indicator, closing **เครื่องหมายปิดยกเลิก** #@]

151 carried-number indicator for addition (varying in length) เครื่องหมายทดกันบวก ._. 77777	156 italic indicator เครื่องหมายแสดงตัวเอียง ?: .
152 directly-over indicator เครื่องหมายแสดงการอยู่ด้านบน ?^ <	157 italic type-form indicator, opening เครื่องหมายแสดงตัวเอียงเปิด :(,'.
153 directly-under indicator เครื่องหมายแสดงการอยู่ด้านล่าง ?v %	158 italic type-form indicator, closing เครื่องหมายแสดงตัวเอียงปิด): .,'
154 termination indicator เครื่องหมายจบสัญลักษณ์ ?]]	159 bold type-form indicator, opening เครื่องหมายแสดงตัวหนาเปิด _(,'_
155 general reference indicator เครื่องหมายอ้างอิงทั่วไป ?-? @]	160 bold type-form indicator, closing เครื่องหมายแสดงตัวหนาปิด)_ _,'

161 percent sign เครื่องหมายเปอร์เซ็นต์ % @0	166 omission sign, general เครื่องหมายยกเว้น ?no =
162 dollar sign เครื่องหมายดอลลาร์ $ @s	167 radical เครื่องหมายราก ^[>
163 cent sign เซ็น ?c @c	168 first inner radical indicator เครื่องหมายรากระดับที่หนึ่ง ^* .
164 pound sterling เครื่องหมายปอนด์ ?l @l	169 second inner radical indicator เครื่องหมายรากระดับที่สอง ^** ..
165 degree sign เครื่องหมายองศา .dg ^.* 	170 index of radical เครื่องหมายระบุรากที่ ^# <

171 asterisk ดอกจัน ?* @#	176 tally เครื่องหมายขีดนับ #_ _
172 crosshatch, pounds (weight) เครื่องหมายช๊าบ ?# .#	177 check mark เครื่องหมายถูก '/ @>
173 caret แคลอทชี้ขึ้น _^ _<	178 ditto mark เครื่องหมายซ้ำ '. ,'
174 inverted caret แคลอทชี้ลง _v _%	179 infinity อนันต์ (ไม่มีที่สิ้นสุด) ..* ,=
175 prime ไพรม์ ?' ,	180 at-sign แอ๊ด @ @a

181 empty or null set เซ็ตว่าง {0} _0	186 integral sign, with superposed circle อินทิเกริล กับวงกลม @tc !@$c]
182 factorial แฟคทอรเรียล & &	187 angstrom unit หน่วยวัดแองสตรอม @as @,a
183 partial derivative พาล์เทลดิริเวอทีป @d @d	188 since (because) เพราะว่า @v @/
184 integral sign อินทิเกริล @t !	189 therefore เพราะฉะนั้น @/ ,*
185 double integral sign อินทิเกริล คู่ @@t !!	190 it does not follow that ไม่เป็นไปตามนั้น /@^ /,*

191 existential quantifier (there exists exactly one) **จำนวน, ขนาด** @eq @=\	
192 universal quantifier **ยูนิเวอร์ซอล คอทิไฟเออร์** @uq @&	

** End of list

3.3 Development of the search table

After the complete list of math symbols with text-based design was compiled, the same table was used to create the search table before writing the source code. All 192 “Text-based print code” along with their corresponding “Braille Nemeth code” were rearranged according to groups from large to small. This arrangement was necessary for the source code to use by linear search. Following is the complete list of the search table.

** Begin of list

065 double shaft, double-pointing arrow **ลูกศรซ้าย ขวา สองเท่า** -hh<> $[77o	054 reverse inclusion with equal sign under **เครื่องหมายเส้นด้านกลับ เท่ากับด้านล่าง** /=&= _.1.k
059 double shaft , right - pointing arrow **ลูกศรชี้ขวาสองเท่า** -hh> $77O	111 ellipse **รูปวงรี** !e $e
061 double shaft , left - pointing arrow **ลูกศรชี้ซ้าย สองเท่า** -hh< $[77	179 infinity **เครื่องหมายอนันต์ (ไม่มีที่สิ้นสุด)** !f ,=
064 ordinary shaft, double-pointing arrow **ลูกศรซ้าย ขวา** -h<> $[33O	165 degree sign **เครื่องหมายองศา** .dg ^.*
053 reverse inclusion with bar under **เครื่องหมายประกอบด้วย เส้นด้านกลับ** /=&- _.1:	185 double integral sign **อินทิเกริล คู่** @@t !!

186 integral sign, with superposed circle **อินทิเกริล กับวงกลม** @tc !@$c]	191 Existential quantifier (there exists exactly one) **จำนวน ขนาด** @eq @=\
187 Angstrom unit **หน่วยวัดแองสตรอม** @as @,a	192 Universal quantifier **ยูนิเวอร์ซอล ควอนทิไฟเออร์** @uq @&
188 Since (because) **เครื่องหมายเพราะว่า** @v @/	181 empty or null set **เซ็ตว่าง** {0} _0
189 Therefore **เครื่องหมายเพราะฉะนั้น** @^ ,*	166 omission sign, general **เครื่องหมายยกเว้น** ?no =
190 It does not follow that **เครื่องหมาย ไม่เป็นไปตามนั้น** /@^ /,*	155 general reference indicator **เครื่องหมายอ้างอิงทั่วไป** ?-? @]

151 carried-number indicator for addition (varying in length) **เครื่องหมายทดการบวก** ._. 77777	139 psi **พซาย** 'ps .y
123 eta **เอต้า** 'et .:	140 omega **โอเมกา** 'om .w
124 theta **ธีตา** 'th .?	115 circle with interior plus sign **เครื่องหมายบวกภายในวงกลม** !c+ $c_$+]
137 phi **ฟาย** 'ph .f	116 circle with interior dot **จุดภายในวงกลม** !c. $c_$*]
138 chi **คาย** 'ch .&	113 right angle **มุมฉาก** ![r $[.r]

098 complex fraction line, diagonal **เส้นแสดงเศษส่วนซ้อน เอียง** @|/ ,_/	069 up-pointing followed by down-pointing arrow **ลูกศรชี้ขึ้น ลูกศรชี้ลง** ^v- $<33o"$%33o
104 right triangle **รูปสามเหลี่ยมมุมฉาก** !tr $t .r]	070 arrow pointing up and down **ลูกศรขึ้นและลง** -^v $<[33O
066 right-pointing over left-pointing arrow **ลูกศรชี้ขวา อยู่เหนือลูกศรชี้ซ้าย** ->< $33O$[33	058 ordinary shaft , right - pointing arrow **ลูกศรชี้ขวาปกติ** -h> $33O
067 left-pointing over right - pointing arrow **ลูกศรชี้ซ้าย อยู่เหนือลูกศรชี้ขวา** -<> $[33$33O	060 ordinary shaft , left- pointing arrow **ลูกศรชี้ซ้ายปกติ** -h< $[33
068 down-pointing followed by up-pointing arrow **ลูกศรชี้ลง ลูกศรชี้ขึ้น** -v^ $%33O"$<33o	050 reverse inclusion **ไม่เป็นสับเซตของ** /=& _.1

051 inclusion with bar under **เครื่องหมายประกอบด้วย** =&- _"k:	045 reverse membership **สมาชิกผกผัน** /&e @5
052 inclusion with equal sign under **เครื่องหมายประกอบด้วย เท่ากับด้านล่าง** =&= _"k.k	036 not perpendicularto **ไม่ตั้งฉาก** /|_ /$p
055 intersection with bar under **เครื่องหมายอินเตอร์เซ็กชั่นมีเส้นด้านล่าง** *:- .%:	037 not parallel to **ไม่ขนาน** /|= /$l
056 union with bar under **เครื่องหมายยูเนียนมีเส้นด้านล่าง** *+- .+:	007 division, spatial **เครื่องหมายหาร ว่าง** -/- 333
047 identity **สมนัยกัน** &id _|	009 plus or minus **เครื่องหมายบวกหรือลบ** +|- +-

010 minus or plus **เครื่องหมายลบหรือบวก** -|+ -+	006 division linear **เครื่องหมายหารยาว** -/ O
011 plus followed by minus **เครื่องหมายบวกและลบ** +&- +"-	005 division **เครื่องหมายหาร** // ./
012 minus followed by plus **เครื่องหมายลบและบวก** -&+ -"+	003 multiplication x **เครื่องหมายคูณเอ็กซ์** ** @*
013 minus followed by minus **เครื่องหมายลบและลบ** -&- -"-	014 intersection **อินเตอร์เซ็กชั่น** *: .%
114 bar horizontal, modifying **เส้นแนวนอน** !- :	015 union **ยูเนี่ยน** *+ .+

017 logical sum **โลจิคอล ผลรวม** @ls @+	028 greater than with bar under **เครื่องหมายมากกว่าหรือเท่ากับ** >- .1:
018 logical product **โลจิคอล ผลคูณ** @lp @%	029 greater than with equal sign under **เครื่องหมายมากกว่าหรือเท่ากับ (ด้านล่าง)** >= .1.k
021 hollow dot **จุดกรวง** .o .*	030 Unequal **เครื่องหมายไม่เท่ากับ** /= /.k
026 less than with bar under **เครื่องหมายน้อยกว่าหรือเท่ากับ** <- "k:	031 less than followed by greater than **เครื่องหมายน้อยกว่าและมากกว่า** <> "k".1
027 less than with equal sign under **เครื่องหมายน้อยกว่าหรือเท่ากับ (ด้านล่าง)** <= "k.k	032 greater than followed by less than **เครื่องหมายมากกว่าและน้อยกว่า** >< .1""k

033 equal sign with question mark over **เครื่องหมายเท่ากับและเครื่องหมายคำถาม** =? ".k<_8]	041 tilde with equal sign under **เท่ากันทุกประการ** ~= @:.k
034 perpendicular to **ตั้งฉาก** |_ $p	102 circle **รูปวงกลม** !c $c
035 parallel to **ขนาน** |= $l	044 is an element of **เป็นสมาชิกของ** &e @e
039 proportion **สัดส่วน** :: ;2	046 equivalence **จำนวนเท่ากัน** == @<,<
040 double tilde **โดยประมาณ** ~~ @:@:	048 variation **แปรผัน** =* _=

<table>
<tr><td>049
inclusion
เป็นสับเซตของ
=&
_"k</td><td>078
angle bracket, right
วงเล็บมุมปิด
]&
..)</td></tr>
<tr><td>057
contracted form of right - pointing arrow
ลูกศรชี้ขวา ขนาดเล็ก
->
$O</td><td>079
vertical bar, double
เส้นตั้งตรงสองเส้น
||
\\</td></tr>
<tr><td>062
arrow pointing up
ลูกศรชี้ขึ้น
|^
$<33O</td><td>080
enlarged parenthesis, left
วงเล็บใหญ่เปิด
*(
,(</td></tr>
<tr><td>063
arrow pointing down
ลูกศรชี้ลง
|v
$%33O</td><td>081
enlarged parenthesis, right
วงเล็บใหญ่ปิด
)*
,)</td></tr>
<tr><td>077
angle bracket, left
วงเล็บมุมเปิด
&[
..(</td><td>082
enlarged bracket, left
วงเล็บก้ามปูใหญ่เปิด
*[
@,(</td></tr>
</table>

083 enlarged bracket, right **วงเล็บก้ามปูใหญ่ปิด**]* @,)	089 simple fraction indicator , opening **เครื่องหมายเปิดเศษส่วน** @(?
084 enlarged brace, left **วงเล็บปีกกาใหญ่เปิด** *{ .,(	092 simple fraction line , diagonal **เส้นแสดงเศษส่วนเอียง** |/ _/
085 enlarged brace, right **วงเล็บปีกกาใหญ่ปิด** }* .,)	093 fractional part of mixed number indicator, opening **เปิดเศษส่วนคละ** @[_?
086 enlarged vertical bar, left and right **เส้นตรงหนา ซ้ายและขวา** *| ,\	094 fractional part of mixed number indicator, closing **ปิดเศษส่วนคละ**]@ _#
088 transcriber's grouping symbol , opening & closing **สัญลักษณ์เปิดปิดกลุ่ม** " ,'	095 complex fraction indicator, opening **เปิดเศษส่วนซ้อน** @{ ,?

096 complex fraction indicator, closing **ปิดเศษส่วนซ้อน** }@ ,#	103 triangle **รูปสามเหลี่ยม** !t $t
097 complex fraction line, horizontal **เส้นแสดงเศษส่วนซ้อน** @/ ,/	105 square **รูปสี่เหลี่ยมจัตุรัส** !s $4
099 shape indicator, regular **เครื่องหมายแสดงรูปทรง ปกติ** ?r $	106 rectangle **รูปสี่เหลี่ยมผืนผ้า** !r $r
100 shaded shape indicator (bold) **เครื่องหมายแสดงรูปทรง หนา** ?b .	107 diamond **รูปสี่เหลี่ยมขนมเปียกปูน** !d $d
101 filled- in shape indicator **เครื่องหมายแสดงรูปทรง ทึบ** ?f _	108 pentagon **รูปห้าเหลี่ยม** !p $5

109 hexagon **รูปหกเหลี่ยม** !h $6	117 alpha **แอลฟ้า** 'a .a
110 star shape **รูปดาว** ?* $s	118 beta **เบตา** 'b .b
112 angle **มุม** ![$[	119 gamma **แกมมา** 'g .g
042 arc, concave upward **เส้นโค้งคว่ำ** !^ $a	120 delta **เดลตา** 'd .d
043 arc, concave downward **เส้นโค้งหงาย** !v $'	121 epsilon **เอปซิลอน** 'e .e

122 zeta เซตตา 'z .z	129 nu นิว 'n .n
125 iota1 ไอโอตา 'i .i	130 xi ซาย 'x .x
126 kappa แคปปา 'k .k	131 omicron โอไมครอน 'o .o
127 lambda แลมดา 'l .l	132 pi พาย 'p .p
128 mu มิว 'm .m	133 rho โร 'r .r

134 sigma **ซิกมา** 's .s	143 letter sign **เครื่องหมายนำตัวอักษร** ?; ;
135 tau **ทาว** 't .t	144 punctuation indicator **เครื่องหมายนำสัญลักษณ์** ?_ _
136 upsilon **อุปซิลอน** 'u .u	147 comma or comma-space within subscript or superscript **เครื่องหมายจุลภาคภายในยกกำลังหรือห้อย** _, [
141 sampi **แซมพาย** 'c .c	149 cancellation indicator, opening **เครื่องหมายเปิดยกเลิก** @# [
142 numeric indicator **เครื่องหมายนำเลข** ?# #	150 cancellation indicator, closing **เครื่องหมายปิด ยกเลิก** #@]

152 directly-over indicator **เครื่องหมายแสดงการอยู่ด้านบน** ?^ <	158 italic type-form indicator, closing **เครื่องหมายแสดงตัวเอียงปิด**): .,'
153 directly-under indicator **เครื่องหมายแสดงการอยู่ด้านล่าง** ?v %	159 bold type-form indicator, opening **เครื่องแสดงตัวหนา เปิด** _(,'_
154 termination indicator **เครื่องหมายจบสัญลักษณ์** ?]]	160 bold type-form indicator, closing **เครื่องหมายแสดงตัวหนา ปิด**)_ _,'
156 italic indicator **เครื่องหมายแสดงตัวเอียง** ?: .	163 cent sign **เครื่องหมายเซ็น** ?c @c
157 italic type-form indicator, opening **เครื่องหมายแสดงตัวเอียงเปิด** :(,'.	164 pound sterling **เครื่องหมายปอนด์** ?l @l

167 radical **เครื่องหมายราก** ^[>	172 Crosshatch pounds (weight) **เครื่องหมายช้าบ** ?# .#
168 First inner radical indicator **เครื่องหมายราก ระดับที่ 1** ^* .	173 Caret **แคลอทชี้ขึ้น** _^ _<
169 second inner radical indicator **เครื่องหมายราก ระดับที่ 2** ^** ..	174 Inverted caret **แคล๊อทชี้ลง** _v _%
170 index of radical **เครื่องหมายระบุรากที่** ^# <	175 prime **ไพรม์** ?' '
171 Asterisk **ดอกจัน** ?* @#	176 Tally **เครื่องหมายขีดนับ** #_ _

177 check mark **เครื่องหมายถูก** '/ @>	019 ampersand **เครื่องหมายแอนด์** & _&
178 ditto mark **เครื่องหมายละ** '. ,'	004 multiplication dot **เครื่องหมายคูณ จุด** * *
087 empty set **เซ็ตว่าง** {} .(.)	001 plus operator **เครื่องหมายบวก** + +
183 partial derivative **พาร์เทลดิริเวทีป** @d @d	002 minus operator **เครื่องหมายลบ** - -
184 integral sign **อินทิเกริล** @t !	022 backslash **เครื่องหมายแบคแสลท** \ _*

<table>
<tr><td>023
equals
เครื่องหมายเท่ากับ
=
.k</td><td>016
tilde
นิเสธ
~
@:</td></tr>
<tr><td>024
less than
เครื่องหมายน้อยกว่า
<
"k</td><td>071
parenthesis, left
เครื่องหมายวงเล็บเปิดเปิด
(
(</td></tr>
<tr><td>025
greater than
เครื่องหมายมากกว่า
>
.1</td><td>072
parenthesis, right
เครื่องหมายวงเล็บปิด
)
)</td></tr>
<tr><td>038
ratio
อัตราส่วน
:
"1</td><td>073
bracket, left
วงเล็บก้ามปูเปิด
[
@(</td></tr>
<tr><td>020
vertical bar
โดยที่
|
\</td><td>074
bracket, right
วงเล็บก้ามปูปิด
]
@)</td></tr>
</table>

075 brace, left **วงเล็บปีกกาเปิด** { .(	146 subscript indicator **เครื่องหมายห้อย** ; ;
076 brace, right **วงเล็บปีกกาปิด** } .)	148 baseline indicator **เครื่องหมายนำแถวปกติ** " "
091 simple fraction line, horizontal **เส้นแสดงเศษส่วน** / /	161 percent sign **เครื่องหมายเปอร์เซ็นต์** % @0
008 division slash **ทับ** / /	162 dollar sign **เครื่องหมายดอลลาร์** $ @s
145 superscript indicator **เครื่องหมายยกกำลัง** ^ ^	180 at-sign **เครื่องหมายแอ้ด** @ @a

182 factorial **แฟคทอรเรียล** & &	

** End of list

3.4 Development of the source code

The text-based math Braille translation engine was developed as a program using Pascal language. The program would act as a function that can be called from the main software. Text-based math string would be passed to the engined for analysis. The engine would loop through the string and **linear search technique** based on the table would be performed at each position of the string. If a match is found the Braille Nemeth code would be substituted. After the entired string was processed, the result would be sent back to the main software.

The main software had to be modified to interface with the engine. Whenever the software found a math string marked by begin math code ## and end math code $$, that portion would be extracted and the engine would be called. The result from the engine in US computer Braillde code was then replaced at the same location of the document. Finally, proper formatting for Braille line length and page length along with Thai word break analysis and Braille page numbering would be performed to produce out put Braille file. The main software used in this research was RS Braille software. Following is the source code of this research.

** Begin of source code

```
program translate_forward_math; {tf-mat.pas}
{
Wiraman Niyomphol
perform math forward translation
```

```
each line of input file is print typing code for math symbols
returns output file with braille math code
use linear search through table dictionary
}

const
brl_id = chr(0);

type
word_array = array [1..200] of string [15];
linestring = string [250];

var
infile, outfile: text;
fn: linestring;
math_p, math_b: word_array;
total_math, ln: integer;
pline, bline: linestring;

procedure load_table;

{load from file tf-mat.tab
data return in array math_p and math_b
insert braille indicator in front of each braille character on math_b
integer total_math is number of entries
format:
skip all heading lines until find a line with *
read 6 lines for each entrey
line 1 is numeric key
line 2 is english description
line 3 is thai description
line 4 is print typing code
```

```
line 5 is computer braille code
line 6 is blank except last entry
if line 6 is * then last entry is found
}

var
infile: text;
i, j: integer;
tline: linestring; {table line}

begin
assign(infile, 'tf-mat.tab');
reset(infile);

{skip heading lines until find a line with *}
tline := '.'; {make sure it is not *}
repeat
readln(infile, tline);
until tline[1] = '*';

i := 0;
tline := '.'; {make sure it is not *}
while (not eof(infile))
and (tline[1] <> '*') do
begin
i := i+1;

{skip numeric key, english description, thai description}
readln(infile, tline);
readln(infile, tline);
readln(infile, tline);
```

```
{get print typing code}
math_p[i] := '';
readln(infile, math_p[i]);

{get computer braille code and add braille identifyer}
readln(infile, tline);
math_b[i] := '';
for j := 1 to length(tline) do
math_b[i] := math_b[i] + brl_id + tline[j];

tline := '.'; {make sure this var is not * before getting line 6}
readln(infile, tline);
{debug:
writeln(i);
writeln(math_p[i], '|');
writeln(math_b[i], '|');
writeln(tline, '|');
readln;
}

end; {while}

{number of entries for science table}
total_math  := i;
if paramstr(1) = '?' then
writeln('Total table entries for science table = ', total_math);
close(infile);
end; {load_table}

function tf_math(var line: linestring): linestring;
{convert string according to math table}
```

```
var
w, p, l: integer;
old_line, temp: linestring;

begin
{empty string, do not process}
if line = '' then
begin
tf_math := '';
exit;
end;

{save original line to return back}
old_line := line;

p := 0;
{loop according to each table enctry}
for w := 1 to total_math do
begin
{keep result in temp during processing}
temp := '';

{find match of this print word entry in line parameter}
p := pos(math_p[w], line);

{if match not found then give whole line to temp}
if p = 0 then
temp := line;

{in case a match is found then keep processing through the line}

{both line and temp will hav mixed converted braile and print code,
```

```
 going through the loop}
while p <> 0 do
begin
{move substring from begin line to match location to keep in temp}
if p > 1 then
temp := temp + copy(line, 1, p-1);

{look back 1 char for braille identifyer,
if found then this current position is braille character already translated,
move it to temp, do not convert again}
if (p > 1) and (line[p-1] = brl_id) then
begin
temp := temp + line[p];
writeln(temp);
end

else
{otherwise add braille word of that match to temp}
temp := temp + math_b[w];

{then delete that print word or braille code from line string}
l := length(math_p[w]);
delete(line, 1, p-1+l);

{continue to search for the same print word in the remaining line string}
p := pos(math_p[w], line);
{if not found then add remaining line string to temp, also jump out of loop}
if p = 0 then
temp := temp + line;

{if found another match of same print word then go back to top of loop}
end; {while}
```

```
{result in temp where all occurrences of this print word are converted}

{give temp back to line to check for another print word}
line := temp;
end; {for}
{all print words are processed and converted if found matches}
{tem has final result}

{insert number sign at beginning of number}
line := temp;
temp := '';
for p := 1 to length(line) do
begin
if (line[p] in ['0'..'9'])
 and (line[p-1] <> brl_id)
 and ((line[p-1] = ' ') or (p = 1)) then
temp := temp + '#' + line[p]
else
temp := temp + line[p];
end;

{convert capital letter for alphabet}
line := temp;
temp := '';
for p := 1 to length(line) do
begin
if (line[p] in ['A'..'Z']) and (line[p-1] <> brl_id) then
temp := temp + ',' + line[p]
else
temp := temp + line[p];
end;
```

```
{strip all braill indentifyer from temp
and give temp back to line}
line := '';
for p := 1 to length(temp) do
begin
        if temp[p] <> brl_id then
line := line + temp[p];
end;

{return converted string as function result}
tf_math := line;
writeln;
writeln(line);
writeln;
{return line parameter with original value}
line := old_line;
      end; {function tf_mat}

begin {main}
if paramstr(1) = '?' then
writeln('Welcom to forward Braille translation engine for mathematical notations');
{get table from tf-mat.tab into arrays}
load_table;

{ask for input and output file
take either keyboard input or command line parameter
if parameter 1 is ? then show messages
}if (paramcount = 0) or (paramstr(1) = '?') then
begin
write('input file: ');
readln(fn);
end
```

```
else
fn := paramstr(1);
assign(infile, fn);
reset(infile);

if (paramcount = 0) or (paramstr(1) = '?') then
begin
write('output file: ');
readln(fn);
end
else
fn := paramstr(2);
assign(outfile, fn);
rewrite(outfile);

{go through the input file line by line}
ln := 0;
while not eof(infile) do
begin
ln := ln+1;
{get the line}
readln(infile, pline);

{check if begin with ", do not translate}
if pline[1] = '"' then
begin
delete (pline, 1, 1); {take out "}
{also take out one space after " if found}
if pline[1] = ' ' then
delete(pline, 1, 1);
bline := pline;
end
```

```
else
{convert according to math table through function call}
bline := tf_math(pline);

{save result to output file}
writeln(outfile, bline);
end; {while}
{finish going through the file}

close(infile);
close(outfile);

if paramstr(1) = '?' then
begin
writeln('finish math forward translation engine');
writeln('by Wiraman Niyompho, October 18, 2006');
end;
end. {translate_forward_mat}

** End of source code
```

3.5 Performance Test

This section looked at both the performance of the software and the ability of the participants to analyze print math expressions and type them properly during the exercise. The Thai Braille translation software (RS Braille) with additional text-based math Braille translation capability was tested with 10 participants at Christian Foundation for the Blind. The participants were math teachers and material production staff from different educational centers in the country. There were 7 sighted persons, 2 low vision persons and 1 blind person. Initially, a three-hour training of how to use the software was given to the participants. Then a two-hour exercise session of 20 math expressions (N1 – N20) developed by the researcher and 3 instances of text with math formulas (Q1 – Q3) extracted from a Thai math book were given to the participants (Basic Mathematics for 10th grade - Book one, 2005). Math expressions (N1 - N20) were tested for its reliability with four Braille experts who are experienced using NEMETH Braille code to write mathematics. They read math expressions in print and used Perkins braillewriter to type out math expressions in Braille manually. Out of 20 expressions (20 full marks), the average correct result was 18.50 or 92.5 percent which is considered acceptable for use as the instrument in the research. The entire training and exercise section took place at the computer lab of the organization. Each participant had a workstation with RS Braille software installed. Visually impaired participants were equipped with appropriate screen reader and screen magnification software.

3.6 Attitude Test

Participants completed a questionnaire consisting of general information, satisfaction level and additional comments. Satisfaction level was measured by asking about overall satisfaction of text-based math input method and overall satisfaction of RS Braille software. The scale was from 1 to 5 (poor, fair, acceptable, good, very good) with following range: Poor 1 – 1.5, Fair 1.6 – 2.5 , Acceptable 2.6 – 3.5, Good 3.6 – 4.5, Very good 4.6 – 5.

In addition, five participants were randomly selected for a semi-structured interview for twenty minutes for their opinion and attitude of text-based method of typing math expressions as well as benefits in supporting mathematical learning for children with visual impairment. The questionnaire took place right after the exercise by filling out questionnaire form on the computer. Follow-up interviews were recorded and transcribed later.

CHAPTER IV
RESULTS

Results of this research will focus on (1) Software Development (2) Performance Test (3) Analysis of 20 math expressions (4) Analysis of 3 math questions and (5) Attitude Test.

4.1 Software Development

As part of the research, a comprehensive list of math symbols based on Nemeth standard was compiled. All 192 items is shown in Appendix A. Researcher designed "Print Typing Code" for all 192 items. For the exercise with 10 participants, 32 items were used and are shown in Table 4.1.

During software development stage, the software was tested by running sample document with all 192 math symbols and all of them translated correctly when appeared individually separated by either a space or a new line. However, some translation errors occurred when math symbols were written together. Attempts were made to correct the software by modifying the source code and rearranging the look-up table. When focusing on math symbols needed for the exercises, two errors were located and the participants were informed during the training and before doing the exercise. The first error involved enlarged parenthesis. Both open and closed enlarged parenthesis did not translate correctly. The participants were told to use regular parenthesis instead. The second error involved fraction and radical. When placing open simple fraction symbol with open square root symbol, this combination did not translate correctly. The participants were told to put a space between them. These temporary solutions did not change the meaning of math questions.

Table 4.1 Math Table with Print typing code and Braille representations (32 items for the exercise)

Item	Symbol	Description	Print typing code	Braille representation
001	+	Plus Operator	+	⠬
002	−	Minus Operator	-	⠤
003	×	Multiplication ×	**	⠈⠡
004	•	multiplication dot	*	⠡
005	÷	Division	//	⠠⠌
023	=	Equals	=	⠨⠅
024	<	less than	<	⠐⠅
025	>	greater than	>	⠨⠂
026	≤	less than with bar under	<-	⠐⠅ ⠱
028	≥	greater than with bar under	>-	⠨⠂ ⠱
030	≠	Unequal	/=	⠌ ⠨⠅
044	∈	is an element of	&e	⠈⠑
071	(	parenthesis, left	(	⠷
072	)	parenthesis, right	)	⠾
075	{	brace, left	{	⠨⠷
076	}	brace, right	}	⠨⠾
089		simple fraction indicator, opening	@(	⠹
090		simple fraction indicator, closing	)@	⠼
091	___	simple fraction line, horizontal	/	⠌
093		fractional part of mixed number indicator, opening	@[	⠸⠹

Table 4.1 (Cont.) Math Table with Print typing code and Braille representations (32 items for the exercise)

Item	Symbol	Description	Print typing code	Braille representation
094		fractional part of mixed number indicator, closing	]@	⠸⠼

The software was tested heavily by using the exercise as the input. Table 4.2, 4.3, and 4.4 show the content of exercise before translation, result of inserting Text-based coded math symbols design in this research and result of translating coded text to Braille according to Nemeth standard. Each table has following details.

Table 4.2 is Print representation, verbal description and Text-based coded math symbols of 20 Math expressions (N1 – N20). The left column shows math expressions visually by using tools in Microsoft Word where the content may not be easily understood when reading with the screen reader. In order to clearify the content of the exercise and provide better accessibility to screen reader user, the middle column was created to give verbal description of 20 math expressions where numbers, variables, decimal points and commas are shown normally.

The heart of this research is shown in the right column :Text-based coded math symbols of 20 math expressions. Notice that begin math marks ## and end math marks $$ are inserted around the math formulas which would be sent to the math translation engine to process and the result will be in Nemeth Braille code. While the question numbers 1 – 20 at the beginning of each line will be translated as “higher number” as usual, numbers in those math formulas will be translated to “lower number” according to Nemeth standard.

Similarly, Table 4.3 shows three columns of print representation, verbal description and Text-based coded math symbols of three questions (Q1 - Q3. Each question was extracted from a Thai math textbook where regular text is written with math expressions (Thailand Ministry of Education, 2004). Each occurrence of a math expression would have to be enclosed by begin math mark ## and end math mark $$. While Thai text was translated to English for this report, the participants typed text portion of the exercise in Thai language.

Table 4.4 shows Braille Translated results in two formats. While the content is the same, the left column is the Translated result shown in US computer Braille and the right column is Translated result shown in Braille font according to US computer Braille. One with knowledge of computer Braille or math Braille should be able to follow along quite easily. For example, math expression N9 with simple fractions content of “9. fraction of 1 over 2 plus fraction of 3 over 4” would be typed as 9. ## @(1/2)@+@(3/4)@ $$ and translated to

(Computer Braille) #i4 ?1/2#+?3/4#

(Braille font) ⠼⠊⠲ ⠹⠂⠌⠆⠼⠬⠹⠒⠌⠲⠼

According to item 89 in Table 4.1, characters @((at sign with open parenthesis) is opening simple fraction indicator. Those two characters would be translated to Nemeth code as dots 1-4-5-6. All other characters in this formula can be referenced to Table 4-1 as follows:

Item 090 closing simple fraction indicator (characters)@ converts to dots 3-4-5-6)
Item 091 simple fraction horizontal line (character / converts to dots 3-4)
Item 001 plus operator (character + converts to dots 3-4-6)

After translation, begin and end math marks will be eliminated and will not be shown in Braille. Since lower numbers appear in a fraction symbol, number sign is not needed. Number 1 would be converted to dot 2, number 2 converted to dots 2-3, and so on. However, question number 9 is still converted as higher number (dots 2-4) proceeded by number sign (dots 3-4-5-6) and followed by regular period dots 2-5-6.

Table 4.2 Twenty math expressions (before translation)

No.	Print representation	Verbal description	Text-based coded math symbols
N1	270.35	270.35	## 270.35 $$
N2	21,375	21,375	## 21,375 $$
N3	10+7-3 = 14	10 plus 7 minus 3 equals 14	## 10+7-3 = 14 $$
N4	8 x 5 ÷ 10 = 4	8 times 5 divided by 10 equals 4	## 8**5//10 = 4 $$
N5	750 < 850	750 less than 850	## 750 < 850 $$
N6	1,300,000 > 130,000	1,300,000 greater than 130,000	## 1,300,000 > 130,000 $$
N7	8 x (3 – 1) = 16	8 time open-parenthesis 3 minus 1 close-parenthesis equals 16	## 8**(3-1) = 16 $$
N8	9 ÷ (3+ 1) = 2.25	9 divided by open-parenthesis 3 plus 1 close-parenthesis equals 2.25	## 9//(3+1) = 2.25 $$
N9	$\frac{1}{2}+\frac{3}{4}$	fraction of 1 over 2 plus fraction of 3 over 4	##@(1/2)@+@(3/4)@ $$
N10	$7\frac{4}{5}-3\frac{5}{7}$	mixed fraction of 7 and 4 over 5 minus mixed fraction of 3 and 5 over 7	## 7@[4/5]@-3@[5/7]@ $$
N11	$7^2 \geq 2^3$	7 squared greater than or equals to 2 cubed	## 7^2 >- 2^3 $$
N12	$\sqrt{9} \leq 4$	square root of 9 less than or equals to 4	## ^[9?] <- 4 $$
N13	$x^2 \neq y^3$	x squared is not equal to y cubed	## x^2 /= y^3 $$
N14	-5 < 0 > -2	negative 5 less than 0 greater than negative 2	## -5 < 0 > -2 $$
N15	$\frac{2+\frac{x}{1}}{\frac{y}{3}-\frac{z}{4}}$	complex fraction of 2 plus fraction of x over 1 all-over fraction of y over 3 minus fraction of z over 4	##@{2+@(x/1)@@/@(y/3)@-@(z/4)@}@ $$
N16	πr^2	pi r squared	## 'pr^2 $$
N17	x^3+2x-1	x cubed plus 2 x minus 1	## x^3"+2x-1 $$

Table 4.2 (Cont.) Twenty math expressions (before translation)

N18	$\frac{\sqrt{x}}{3}$	fraction of square root of x over 3	## @(^[x?]/3)@ $$
N19	$\sqrt[3]{27} = 3$	cube root of 27 equals 3	## ^#3^[27?] = 3 $$
N20	$x_1 + x_2 < x_n$	x sub 1 plus x sub 2 less than x sub n	## x1+x2 < x;n $$

Table 4.3 Three math questions consisting of text and expressions (before translation)

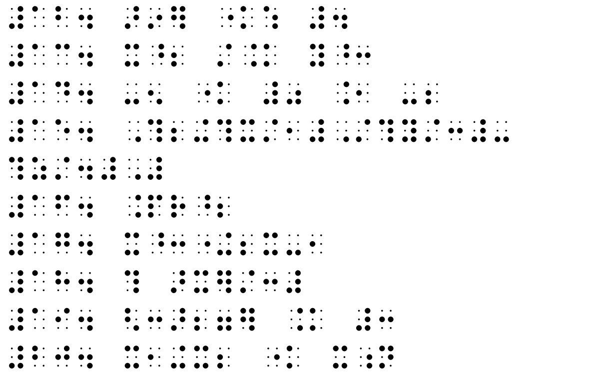
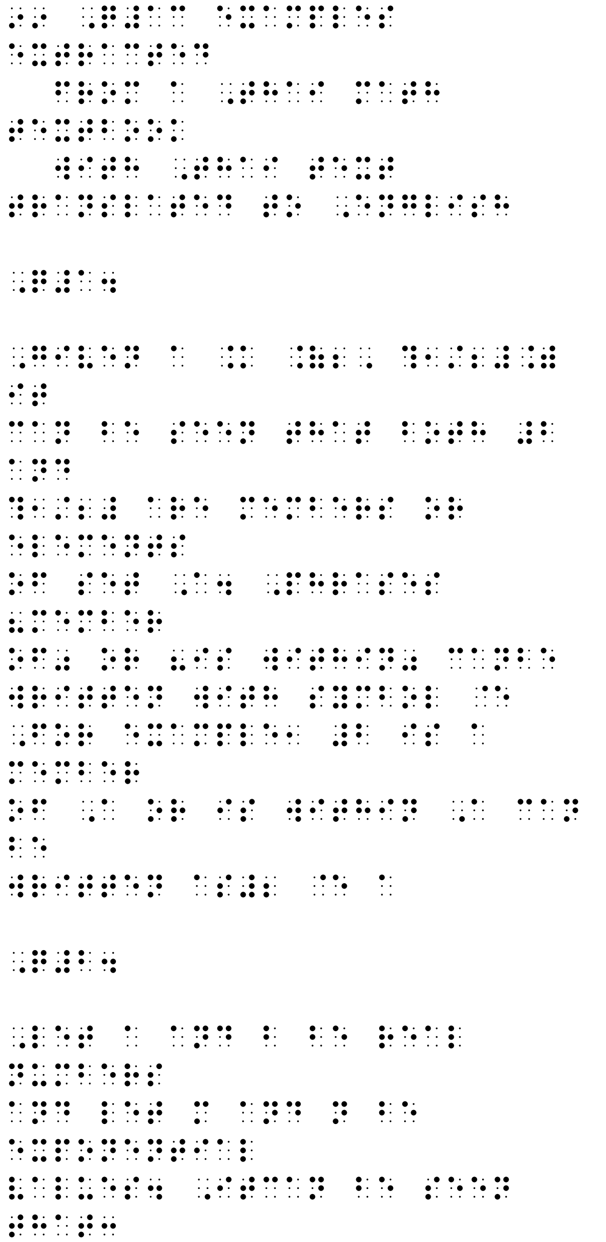
No.	**Print representation**	**Verbal description**	**Text-based coded math symbols**
Q1	Given A = {2, $\frac{1}{2}$}, it can be seen that both 2 and $\frac{1}{2}$ are members or elements of set A. Phrases "member of" or "is within" can be written with symbol $\in$. For example, 2 is a member of A or is within A can be written as $2 \in A$	Given A = {2, $\frac{1}{2}$}, it can be seen that both 2 and $\frac{1}{2}$ are members or elements of set A. Phrases "member of" or "is within" can be written with symbol $\in$. For example, 2 is a member of A or is within A can be written as $2 \in A$	Given ## A = {2, @(1/2)@} $$ it can be seen that both 2 and ## @(1/2)@ $$ are members or elements of set A. Phrases "member of" or "is within" can be written with symbol ## &e $$ For example, 2 is a member of A or is within A can be written as ## 2 &e A $$
Q2	Let a and b be real numbers and let m and n be exponential values. It can be seen that: 1. $a^m \times a^n = a^{m+n}$ 2. $a^m \times b^m = (a \times b)^m$ 3. $(a^m)^n = a^{mn}$ 4. $a^m \div a^n = a^{m-n}$, a, n $\neq 0$	1. a to the power of m times a to the power of n equals a to the power of m plus n 2. a to the power of m times b to the power of m equals open-parenthesis a times b close-parenthesis to the power of m 3. open-parenthesis a to the power of m close-parenthesis to the power of n equals a to the power of m n 4. a to the power of m divided by a to the power of n equals to a to	Let a and b be real numbers and let m and n be exponential values. It can be seen that: 1. ## a^m**a^n = a^m+n $$ 2. ## a^m**b^m = (a**b)^m $$ 3. ## (a^m)^n = a^mn $$ 4. ## a^m//a^n = a^m-n, a /= 0 $$ 5. ## @(a^n/b^n)@ = (@(a/b)@)^n, b /= 0 $$

No.	Print representation	Verbal description	Text-based coded math symbols
	5. $\frac{a^n}{b^n} = (\frac{a}{b})^n$, $b \neq 0$ Example: 1. $64^{\frac{5}{6}} = ((64)^{\frac{1}{6}})^5$ $= ((2^6)^{\frac{1}{6}})^5$ $= 2^5$	the power of m minus n where a is not equal to 0 5. fraction of a to the power of n over b to the power of n equals open-parenthesis fraction of a over b close-parenthesis to the power of n where b is not equal to 0 Example: 1. 64 to the power of fraction of 5 over 6 equals open-parenthesis open-parenthesis 64 close-parenthesis to the power of fraction of 1 over 6 close-parenthesis to the power of 5 equals open-parenthesis open-parenthesis 2 to the power of 6 close-parenthesis to the power of fraction of 1 over 6 close-parenthesis to the power of 5 equals 2 to the power of 5	Example: 1. ## 64^@(5/6)@ = ((64)^@(1/6)@)^5 $$ ## = ((2^6)^@(1/6)@)^5 $$ ## = 2^5 $$
Q3	Example 13: Solve expression $x^2 < 1$ and show the answer by using line number. Method: From $x^2 < 1$ derives $x^2 – 1 < 0$ and $(x+1)(x-1) < 0$ Consider the value of $(x+1)(x-1)$ in the range of $(-\infty,-1)$, $(-1,1)$ and $(1,\infty)$ by choosing the	Example 13: Solve expression x squared less than 1 and show the answer by using line number. Method: From x squared less than 1 derives x squared minus 1 less than 0 and open-parenthesis x plus 1 close-parenthesis open-parenthesis x minus 1 close-	Example 13: Solve expression ## x^2 < 1 $$ and show the answer by using line number. Method: From ## x^2 < 1 $$ derives ## x^2"-1 < 0 $$ and ## (x+1)(x-1) < 0 $$ Consider the value of ## (x+1)(x-1) $$ in the range of ## (-..*, -1), (-1, 1) $$ and## (1,

No.	Print representation	Verbal description	Text-based coded math symbols
	value of x within such range as follows:	parenthesis less than 0 Consider the value of open-parenthesis x plus 1 close-parenthesis open-parenthesis x minus 1 close-parenthesis in the range of open-parenthesis negative infinity, -1 close-parenthesis, open-parenthesis -1, 1 close-parenthesis and open-parenthesis 1, infinity close-parenthesis by choosing the value of x within such range as follows:	..*) $$ by choosing the value of x within such range as follows:

Table 4.4 Translated result shown in US computer Braille and Braille font

Braille page	Translated result in US computer Braille	Translated result in Braille font
1	Exercises: text-based coded math symbols of 20 math questions and q1-3 * 20 math questions #a4 #270.35 #b4 #21,375 #c4 #10+7-3 .k #14 #d4 #8@*5./10 .k #4 #e4 #750 "k #850 #f4 #1,300,000 .1 #130,000 #g4 #8@*(3-1) .k #16 #h4 #9./(3+1) .k #2.25 #i4 ?1/2#+?3/4# #aj4 #7_?4/5_#-3_?5/7_# #aa4 #7^2 .1: #2^3 #ab4 >9] "k: #4 #ac4 x^2 /.k y^3 #ad4 -5 "k #0 .1 -2	⠠⠑⠭⠑⠗⠉⠊⠎⠑⠎⠒ ⠠⠞⠑⠭⠞⠤⠃⠁⠎⠑⠙ ⠉⠕⠙⠑⠙ ⠍⠁⠞⠓ ⠎⠽⠍⠃⠕⠇⠎ ⠕⠋ ⠼⠃⠚ ⠍⠁⠞⠓ ⠟⠥⠑⠎⠞⠊⠕⠝⠎ ⠁⠝⠙ ⠟⠼⠁⠉ ⠔⠔ ⠼⠃⠚ ⠍⠁⠞⠓ ⠟⠥⠑⠎⠞⠊⠕⠝⠎ ⠼⠁⠲ ⠼⠆⠶⠴⠨⠒⠢ ⠼⠃⠲ ⠼⠆⠂⠠⠒⠶⠢ ⠼⠉⠲ ⠼⠂⠴⠬⠶⠤⠒ ⠨⠅ ⠼⠂⠲ ⠼⠙⠲ ⠼⠦⠈⠡⠢⠨⠌⠂⠴ ⠨⠅ ⠼⠲ ⠼⠑⠲ ⠼⠶⠢⠴ ⠐⠅ ⠼⠦⠢⠴ ⠼⠋⠲ ⠼⠂⠠⠒⠴⠴⠠⠴⠴⠴ ⠨⠂ ⠼⠂⠒⠴⠠⠴⠴⠴ ⠼⠛⠲ ⠼⠦⠈⠡⠷⠒⠤⠂⠾ ⠨⠅ ⠼⠂⠖ ⠼⠓⠲ ⠼⠔⠨⠌⠷⠒⠬⠂⠾ ⠨⠅ ⠼⠆⠨⠆⠢ ⠼⠊⠲ ⠹⠂⠌⠆⠼⠬⠹⠒⠌⠲⠼ ⠼⠁⠚⠲ ⠼⠶⠸⠹⠲⠌⠢⠸⠼⠤⠒⠸⠹⠢⠌⠶⠸⠼ ⠼⠁⠁⠲ ⠼⠶⠘⠆ ⠨⠂⠱ ⠼⠆⠘⠒ ⠼⠁⠃⠲ ⠜⠔⠻ ⠐⠅⠱ ⠼⠲ ⠼⠁⠉⠲ ⠭⠘⠆ ⠌⠨⠅ ⠽⠘⠒ ⠼⠁⠙⠲ ⠤⠢ ⠐⠅ ⠼⠴ ⠨⠂ ⠤⠆

Braille page	Translated result in US computer Braille	Translated result in Braille font
	#ae4 ,?2+?x/1#,/?y/3#- ?z/4#,# #af4 .pr^2 #ag4 x^3"+2x-1 #ah4 ? >x]/3# #ai4 <3>27] .k #3 #bj4 x1+x2 "k x;n	
2	* q1-3 examples extracted from a thai math textbook with thai text translated to english ,q#a4 ,given A .k .(2, ?1/2#.) it can be seen that both #b and ?1/2# are members or elements of set ,a4 ,phrases 8member of0 or 8is within0 canbe written with symbol @e ,for example1 #b is a member of ,a or is within ,a can be written as#2 @e A ,q#b4 ,let a and b be real numbers and let m and n be exponential values4 ,itcan be seen that3 #a4 a^m@*a^n .k a^m+n #b4 a^m@*b^m .k (a@*b)^m	

Braille page	Translated result in US computer Braille	Translated result in Braille font
		⠼⠁⠲ ⠁⠘⠍⠈⠡⠁⠘⠝ ⠨⠅ ⠁⠘⠍⠬⠝ ⠼⠃⠲ ⠁⠘⠍⠈⠡⠃⠘⠍ ⠨⠅ ⠷⠁⠈⠡⠃⠾⠘⠍
3	#c4 (a^m)^n .k a^mn #d4 a^m./a^n .k a^m-n, a /.k #0 #e4 ?a^n/b^n# .k (?a/b#)^n, b /.k #0 ,example3 #a4 #64^?5/6# .k ((64)^?1/6#)^5 .k ((2^6)^?1/6#)^5 .k #2^5 ,q#c4 ,example #ac3 ,solve expression x^2 "k #1 and show the answer by using line number4 ,method3 ,from x^2 "k #1 derives x^2"-1 "k #0 and (x+1)(x- 1) "k	⠼⠉⠲ ⠷⠁⠘⠍⠾⠘⠝ ⠨⠅ ⠁⠘⠍⠝ ⠼⠙⠲ ⠁⠘⠍⠨⠌⠁⠘⠝ ⠨⠅ ⠁⠘⠍⠤⠝⠠ ⠁ ⠌⠨⠅ ⠼⠴ ⠼⠑⠲ ⠹⠁⠘⠝⠌⠃⠘⠝⠼ ⠨⠅ ⠷⠹⠁⠌⠃⠼⠾⠘⠝⠠ ⠃ ⠌⠨⠅ ⠼⠴ ⠠⠑⠭⠁⠍⠏⠇⠑⠒ ⠼⠁⠲ ⠼⠖⠲⠘⠹⠢⠌⠖⠼ ⠨⠅ ⠷⠷⠖⠲⠾⠘⠹⠂⠌⠖⠼⠾⠘⠢ ⠨⠅ ⠷⠷⠆⠘⠖⠾⠘⠹⠂⠌⠖⠼⠾⠘⠢ ⠨⠅ ⠼⠆⠘⠢ ⠠⠟⠼⠉⠲ ⠠⠑⠭⠁⠍⠏⠇⠑ ⠼⠁⠉⠒ ⠠⠎⠕⠇⠧⠑ ⠑⠭⠏⠗⠑⠎⠎⠊⠕⠝ ⠭⠘⠆ ⠐⠅ ⠼⠂ ⠁⠝⠙ ⠎⠓⠕⠺ ⠞⠓⠑ ⠁⠝⠎⠺⠑⠗ ⠃⠽ ⠥⠎⠊⠝⠛ ⠇⠊⠝⠑ ⠝⠥⠍⠃⠑⠗⠲ ⠠⠍⠑⠞⠓⠕⠙⠒ ⠠⠋⠗⠕⠍ ⠭⠘⠆ ⠐⠅ ⠼⠂ ⠙⠑⠗⠊⠧⠑⠎

Braille page	Translated result in US computer Braille	Translated result in Braille font
	#0	⠭⠘⠆⠐⠤⠂ ⠐⠅ ⠼⠴ ⠁⠝⠙ ⠷⠭⠬⠂⠾⠷⠭⠤ ⠂⠾ ⠐⠅ ⠼⠴
4	,consider the value of (x+1)(x-1)in the range of (-,=, -1), (-1, #1) and(1, ,=) by choosing the value of x within such range as follows3	⠠⠉⠕⠝⠎⠊⠙⠑⠗ ⠞⠓⠑ ⠧⠁⠇⠥⠑ ⠕⠋ ⠷⠭⠬⠂⠾⠷⠭⠤⠂⠾⠊⠝ ⠞⠓⠑ ⠗⠁⠝⠛⠑ ⠕⠋ ⠷⠤⠠⠿⠠ ⠤⠂⠾⠠ ⠷⠤⠂⠠ ⠼⠂⠾ ⠁⠝⠙⠷⠂⠠ ⠠⠿⠾ ⠃⠽ ⠉⠓⠕⠕⠎⠊⠝⠛ ⠞⠓⠑ ⠧⠁⠇⠥⠑ ⠕⠋ ⠭ ⠺⠊⠞⠓⠊⠝ ⠎⠥⠉⠓ ⠗⠁⠝⠛⠑ ⠁⠎ ⠋⠕⠇⠇⠕⠺⠎⠒

4.2 Performance Test

During the three-hour training, the participants received introductory instruction of RS Braille software on how to enter text-based math symbols by using begin and end math indicators and focused on mathematical topics used in the exercises without showing actual math expressions of the exercise. It was observed that the participants actively worked on these areas: basic operators, relational operators, simple fraction, mixed fraction, complex fraction, radical, indexed radical, exponential, subscript and baseline indicator. While working on his/her own workstation, the participants asked questions and worked with each other to solve the problem as well. Even though the participants were not required to have knowledge of math Braille, the researcher emphasized the fact that everything had to be typed in linear style from left to right and special attention for proper opening and closing indicators which may not exist in print had to be given. The use of base line indicator was explained in detail with specific examples as the symbol determined when an exponential value or a subscript value was finished.

After the training, all participants worked on the exercises for about 2 hours. They were given print exercises and the task was to type all 20 math expressions (N1 – N20) and 3 questions with text (Q1 - Q3) in RS Braille editor with proper text-based math symbols. One low vision person read print exercises independently. The researcher dictated the exercises to the other low vision person and the blind person. Both print and translated Braille files of all participants were copied and printed out for analysis.

The researcher made indications of full mark for each question and reviewed the performance of the participants and corrected all exercises. For example, question N9 (see Table 4.2) with math expression one half plus three forth would have 9 points consisting of 2 points for begin and end math markers ## and $$, 3 points for the first simple fraction; open, close and fraction line @(/)@, 3 points for the second simple fraction and 1 point for the plus sign +.

As shown in Table 4.5 and Table 4.6, total number of points for the entire exercise is 280 where the first set of 20 math expressions (N1 – N20) is 125 and questions with text (Q1 - Q3) have total points of 20, 85 and 50 respectively. Out of 280 possible points, all participants earned the average of 266.90 points which is equal to 95.32 percent accuracy.

Table 4.5 Exercise Results of 20 math Expressions and 3 Math Questions

Participants													
Q	total	A	B	C	D	E	F	G	H	I	J	Average	%
1	3	3	3	3	3	3	3	3	3	3	3	3.00	100.00
2	3	3	3	3	3	3	3	3	3	3	3	3.00	100.00
3	5	5	5	5	5	5	5	5	5	5	5	5.00	100.00
4	5	4	5	4	3	5	5	5	5	5	5	4.60	92.00
5	3	3	3	3	3	3	3	3	3	3	3	3.00	100.00
6	6	6	6	6	6	6	6	6	6	6	6	6.00	100.00
7	7	7	7	7	6	7	7	7	7	7	7	6.90	98.57
8	8	8	7	8	7	8	7	8	8	8	8	7.70	96.25
9	9	9	9	9	9	9	9	9	9	9	9	9.00	100.00
10	9	9	9	9	5	5	5	5	9	9	9	7.40	82.22
11	5	5	5	5	2	4	4	5	5	5	5	4.50	90.00
12	5	4	5	4	2	3	4	4	3	5	5	3.90	78.00
13	5	5	5	5	5	5	5	5	5	5	5	5.00	100.00
14	6	6	6	6	6	6	6	6	6	6	6	6.00	100.00
15	16	16	16	16	13	12	12	15	15	16	16	14.70	91.88
16	4	4	4	4	4	4	4	4	4	4	4	4.00	100.00
17	6	6	6	6	4	6	6	6	5	6	6	5.70	95.00
18	7	6	7	6	4	5	5	5	5	7	6	5.60	80.00
19	6	6	6	6	3	4	5	4	5	6	6	5.10	85.00
20	7	7	7	7	3	7	6	7	7	7	6	6.40	91.43
total	**125**	**122**	**124**	**122**	**96**	**110**	**110**	**115**	**118**	**125**	**123**	**116.50**	**93.20**
Q1	9	9	9	9	7	9	8	8	9	9	9	8.60	95.56
	5	5	5	5	4	5	4	2	5	5	5	4.50	90.00
	3	3	3	3	1	3	3	3	3	3	3	2.80	93.33
	3	3	3	3	1	3	3	3	3	3	3	2.80	93.33
total	**20**	**20**	**20**	**20**	**13**	**20**	**18**	**16**	**20**	**20**	**20**	**18.70**	**93.50**
Q2	8	8	8	8	7	8	8	8	8	8	8	7.90	98.75
	10	10	10	10	8	10	10	10	10	10	10	9.80	98.00
	8	8	8	8	8	8	8	8	8	8	8	8.00	100.00
	10	10	10	10	10	10	10	10	10	10	10	10.00	100.00
	16	16	16	16	14	14	16	14	16	16	16	15.40	96.25
	16	16	16	16	16	12	16	12	16	16	16	15.20	95.00
	13	13	13	13	13	10	13	8	13	13	13	12.20	93.85
	4	4	4	4	4	4	4	3	4	4	4	3.90	97.50
total	**85**	**85**	**85**	**85**	**80**	**76**	**85**	**73**	**85**	**85**	**85**	**82.40**	**96.94**

Table 4.5 (Cont.) Exercise Results of 20 math Expressions and 3 Math Questions

Participants													
Q	**total**	A	B	C	D	E	F	G	H	I	J	Average	%
Q3	**4**	4	4	4	4	4	4	4	4	4	4	4.00	100.00
	4	4	4	4	4	4	4	4	4	4	4	4.00	100.00
	6	6	6	6	6	6	6	3	5	6	6	5.60	93.33
	9	9	9	9	9	9	9	8	9	9	9	8.90	98.89
	8	8	8	8	8	8	8	6	8	8	8	7.80	97.50
	13	13	13	13	13	13	13	13	13	13	13	13.00	100.00
	6	6	6	6	6	6	6	6	6	6	6	6.00	100.00
total	**50**	**50**	**50**	**50**	**50**	**50**	**50**	**44**	**49**	**50**	**50**	**49.30**	**98.60**
final	**280**	**277**	**279**	**277**	**239**	**256**	**263**	**248**	**272**	**280**	**278**	**266.90**	**95.32**

Table 4.6 Summary of Exercise Results

Item	**Full Mark**	**Earned Average**	**Earned Percentage**
N1-N20	125	116.50	93.20
Q1	20	18.70	93.50
Q2	85	82.40	96.94
Q3	50	49.30	98.60
Final	280	266.90	95.32

4.3 Analysis of 20 math expressions N1 – N20

Since this is the first time for the participants to use this technique and they did not have much practice time, it is understandable that they will be mistakes. Basic result of each questions of all participants were analyzed in details in terms of individual mistakes, typical pattern, formatting and actual Braille code for improving training guidelines and software development in the future. Detailed analysis along with verbal description and text-based coded math symbols will be shown for reference.

N1. 270.35
(coded) 1. ## 270.35 $$
N2. 21,375
(coded) 2. ## 21,375 $$
N3. 10 plus 7 minus 3 equals 14
(coded) 3. ## 10+7-3 = 14 $$

The first three questions are straight forward, numbers with decimal point, comma and some basic and relational operators namely plus sign, minus sign and equal sign. There was no mistakes on these questions. Overall performance for all three questions was 100 percent.

N4. 8 times 5 divided by 10 equals 4
(coded) 4. ## 8**5//10 = 4 $$

This question uses cross or multiplication sign x instead of a dot above the number. That means dots 4 and 1-6 need to be used for the sign and the typist should have typed **. One subject typed * indicating multiplication dot and two subjects typed *x reflecting either typing mistake or lack of understanding of the concept. This question also uses basic division sign where the typist could type // which will convert to dots 4-6 and 3-4. Regular / is reserved for fraction line. One subject chose another division sign which was not suitable for this context. Overall performance on this question was 92 percent.

N5. 750 less than 850
(coded) 5. ## 750 < 850 $$
N6. 1,300,000 greater than 130,000
(coded) 6. ## 1,300,000 > 130,000 $$

These two questions use additional relational operators (less than and greater than). There were no mistakes on N5 and N6 but a questionable situation came up. Two subjects did not put space before and after those relational operators. Typically spaces are recommended before and after the relational operator in order to separate two sides of the equation clearly. However, the lack of spaces is not considered completely incorrect. So far, basic relational operators (equl to, less than, greater than) are used. Overall performance for these two questions was 100 percent.

At this point, most subjects should have had the grasp of text-based method of typing basic math symbols namely + for plus sign, - for minus sign, ** for cross multiplication sign, // for basic division sign, and simple relational operators such as = for equal sign, < for less than sign and > for greater than sign.

N7. 8 time open-parenthesis 3 minus 1 close-parenthesis equals 16
(coded) 7. ## 8**(3-1) = 16 $$

N8. 9 divided by open-parenthesis 3 plus 1 close-parenthesis equals 2.25
(coded) 8. ## 9//(3+1) = 2.25 $$

The focus of questions N7 and N8 is parenthesis. Since open and close parentheses (and) are typical symbols used regularly, all subjects had no problem with parentheses. However, mistakes and questionable situations were in other areas. Some subjects put spaces inappropriately. There were three instances of no space before and after equal sign. Although it is not wrong, it would be less confusing to have space surrounding equal sign. In contrary, it is not necessary to put space within the expression. One subject put space after multiplication sign in N7 and another subject put space after division sign in N8. Actual mistakes were wrong multiplication sign in N7, wrong division sign in N8, using minus instead of plus sign in N8 and missing closing math code $$ in N8. Overall performance was 98.57 percent and 96.25 percent for N7 and N8 respectively.

N9. fraction of 1 over 2 plus fraction of 3 over 4
(coded) 9. ## @(1/2)@+@(3/4)@ $$

This question focuses on simple fraction. The typist had to understand that there are different types of fractions namely simple fraction, mixed fraction and complex fraction. Each type of fraction requires different Braille symbols and therefore requires different text-based math symbols. In addition, open fraction and close fraction symbols need to be typed along with the fraction line. Given open simple fraction symbol is @(and close simpoe fraction symbol is)@ the typist would have to type @(1/2)@ for simple fraction one over two or one half. All subjects typed everything correctly. Only one subject added a space before plus sign and another space after open fraction sign. These extra spaces do not look nice but are not considered mistakes. Overall performance was 100 percent.

N10. mixed fraction of 7 and 4 over 5 minus mixed fraction of 3 and 5 over 7
(coded) 10. ## 7@[4/5]@-3@[5/7]@ $$

As the subjects moved on to mixed fraction, several of them began to get confused and typed wrong open and close fraction symbols. The design here is to use @ (at sign) with three level of parenthesis to indicate the type of fraction. Regular parentheses are used for simple fraction, brackets are used for mixed fraction and braces are used for complex fraction. Mistakes range from typing only at sign without bracket, using at sign with brace instead of bracket and switching the order of at sign and bracket. Overall performance was 82 percent.

N11. 7 squared greater than or equals to 2 cubed
(coded) 11. ## 7^2 >- 2^3 $$

The focus here is exponential sign. All subjects typed ^ caret for the correct sign except one subject used _ underline sign instead. Three subjects misunderstood the use of baseline indicator dot 5. After superscript (exponential), dot 5 should be used to indicate the end of that exponential value. The code for this is quotation mark. However, it is not necessary in N11 because the exponential power is at the end of the expression where space and end of line is involved. These subjects typed 7^2” >- 2^3” where quotes are not needed. Another area of mistake is the use of relational operator “greater than or equals to”. Even though we say “greater than or equals to”, the sign in math is equals sign with a bar under. The code for this is >- but mistakes were >< and >=. Overall performance was 90 percent.

N12. square root of 9 less than or equals to 4
(coded) ## ^[9?] <- 4 $$

The focus here is square root. The subjects had to specify open square root ^[and close square root ?] surround 9. All subjects used correct open square root symbol but two subjects typed closed square root symbol incorrectly. Many subjects still had problems with relational symbol. There were eight instances of mistakes for less than or equals to <- ranging from no space surrounding the symbol, missing dash sign after less than sign, typing equal sign instead of dash sign and wrong combination of less, greater and equal signs. Overall performance was 78 percent.

N13. x squared is not equal to y cubed
(coded) ## x^2 /= y^3 $$

This question tested the use of “not” where slash is applied in front of several symbols in Nemeth code. In this case /= represents “not equal to” sign. All subjects answered this question correctly except two subjects had two questionable situations: no space surround /= and unnecessary use of quote (baseline indicator) after the exponential value. Overall performance was 100 percent.

N14. negative 5 less than 0 greater than negative 2
(coded) ## -5 < 0 > -2 $$

All subjects answered the question correctly. Three subjects did not put space surrounding relational operators. Normally this does not cost any problem but in this case of “greater than negative 2” caused wrong Braille translation. By typing >-2 without space the software took >- for “greater than or equal to” sign and Therefore translated to “greater than or equals to 2” instead of “greater than negative 2”. So the issue of not typing space surround the relational operators is really critical. One way to solve this problem is to enforce strict rule that there must be space surround all relational operators. Another way to solve this problem is to change the symbol for “less than or equals to” and “greater than or equals to” to conform with the name instead of the appearance of the operators. That means an equal sign instead of a dash sign should be used (use <= and >= instead of <- and >- symbols) even though in print the bar is written under the signs. The table would have to be modified for the software for this solution. Even though an important issue arised in this question and the translation was wrong for some subjects, the subjects were not at fault in this experiment. Therefore, overall performance for this questions was 100 percent.

N15. complex fraction of 2 plus fraction of x over 1 all-over fraction of y over 3 minus fraction of z over 4
(coded) ## @{2+@(x/1)@@/@(y/3)@-@(z/4)@}@ $$

As stated earlier, there are three levels of fraction representation in Nemeth code: simple fraction, mixed fraction and complex fraction. In this case, simple fraction and complex fraction are involved. The subject had to analyze the expression and identify first the complex fraction and specify where to use open complex fraction @{, complex fraction line @/ and close complex fraction }@ symbols. These three symbols will translate to dots 6, 1-4-5-6 (open), dots 6, 3-4 (fraction line) and dots 6, 3-4-5-6 (close) accordingly. The main numerator has one simple fraction and the main denominator has two simple fractions. Some subjects specified wrong selection of symbols for different levels of fractions by using @( )@ or @[ ]@ instead of @{ }@ for complex fraction. Complex fraction line also was typed incorrectly and typically the at sign in @/ was missing. This is due to the fact that fraction line for other types (simple fraction and mixed fraction) requires only / symbol. Other mistakes were typing wrong numbers, adding unnecessary spaces and switching the sequence of characters within the symbols. Overall performance was 91.88 percent.

Since at sign and different types of parenthesis were used to deal with fraction representation, subjects semed to be confused regarding what to type and keeping track of various text symbols visually. If the subject added a space between at sign and parenthesis such as @ (or) @ then it would not be translated as open and closed simple fraction and (and) would be translated as parenthesis instead. If actual parenthesis is involved in an expression with fractions then the typist may have difficult time identifying between parentheses and fraction signs. The code of)@(could represent either close parenthesis follows by open simple fraction or close simple fraction follows by open parenthesis. There are two ways to deal with this. The first solution is to insert spaces to help separate symbols and let the software delete spaces later.

(coded with spaces) ## @{ 2+@(x/1)@ @/ @(y/3)@-@(z/4)@ }@ $$

Another solution is to redesign all fraction symbols to begin with at sign. It may not look symmetrical but it would avoid confusion and follows Nemeth code logic. Simple fraction would become @(/ @) and mixed fraction would become @[/ @] and complex fraction would become @{ @/ @} for example.

(coded begin with at sign) ## @{2+@(x/1@)@/@(y/3@)-@(z/4@)@} $$

N16. pi r squared
(coded) ## 'pr^2 $$

This question requires the use of Greek character which is coded by placing ‘ in front of a key letter. For Pi, the code is ‘p which would be translated to dots 4-6, 1-2-3-4. All subjects had no problem with this one and overall performance was 100 percent.

N17. x cubed plus 2 x minus 1
(coded) ## x^3"+2x-1 $$

This question tested the use of quote symbol or base line indicator. By typing quote, it means the exponential value is only 3 and the remaining portion of the expression does not apply to the exponent. Two subjects forgot to put quote. Other mistakes were typing / instead of + and adding unnecessary spaces within the expression. Overall performance was 95 percent.

N18. fraction of square root of x over 3
(coded) ## @(^[x?]/3)@ $$

Due to software error, this question requires a space after open fraction sign and the subjects were informed. Besides knowing how to specify open and close fraction symbols, the subjects had to specify open and close square root as well. Three subjects had problem with square root symbols where bbraces ^{ ?} were used instead of brackets ^[?] and missing ? as part of close square root symbol. One subject forgot to put space after open fraction sign and caused ^[not to translate to open square root sign. Other mistakes were incorrect fraction symbols and adding extra begin math code ##. Overall performance was 80 percent.

N19. cube root of 27 equals 3
(coded) ## ^#3^[27?] = 3 $$

Although the focus of this question is specifying index radical by using ^# as index radical symbol, many subjects still had problems with regular open and close radical symbols. Dealing with radical symbols can be as difficult as fraction symbols. The subjects had to observe the expression and understand that the value under the radical sign would have to be enclosed by open and close radical symbols or square root symbols. If another value is placed above the radical sign as an index value then that value must be placed in front of open radical symbols and preceded by index radical symbol. Mistakes ranged from omitting all symbols due to confusion, using braces instead of brackets, omitting number sign and omitting question mark. Overall performance was 85 percent.

N20. x sub 1 plus x sub 2 less than x sub n
(coded) ## x1+x2 < x;n $$

The focus of this question is the use of subscript symbol. The code for subscript is ; which would translate to dots 5-6. In case of the subscript value is a number such as x subscript 1, Nemeth code allows the use of subscript value right after the base value without having to add semicolon. Therefore, x subscript 1 can be typed as x1 and x subscript n must be typed as x;n. One subject added ; for x1 and x2 and another subject incorrectly used _ instead of ; for subscript symbol. Although it is not necessary, some subjects used baseline indicator after the subscript such as x1"+x2" to indicate the end of subscript value. Other typical mistakes were extra begin math code ##, missing end math code $$ and typing incorrect relational operator. Overall performance was 91.43 percent.

4.4 Analysis of 3 math questions

In remaining part of the exercise Q1, Q2 and Q3 where the subjects were required to type text along with the expressions, the subjects typed text quite accurately. The subjects had to pay attention to when a math symbol or a math expression began and needed to insert begin and end math code ## $$ properly.

Q1. This question contains 4 instances that needed coding. Symbols { and } were used to show a set. Members of the set were regular number and fractions. The element symbol was needed and the subjects had to search the table for the code for this symbol.

Q1.1 ## A = {2, @(1/2)@} $$
Q1.2 ## @(1/2)@ $$
Q1.3 ## &e $$
Q1.4 ## 2 &e A $$

The subjects typed set symbols and located member symbol &e correctly. Mistakes were missing begin math code, missing end math code, incorrect fraction line and missing at sign in close fraction symbol. Overall performance was 93.50 percent.

Q2. This question contains 8 instances that needed coding. Q2.1 through Q2.5 are general rules regarding operations on exponent. Q2.6 through Q2.8 are expressions of an exponential example. In case that enlarge parentheses were used, the subjects were informed to replace with regular parentheses instead.

Q2.1 ## a^m**a^n = a^m+n $$
Q2.2 ## a^m**b^m = (a**b)^m $$
Q2.3 ## (a^m)^n = a^mn $$
Q2.4 ## a^m//a^n = a^m-n, a /= 0 $$
Q2.5 ## @(a^n/b^n)@ = (@(a/b)@)^n, b /= 0 $$
Q2.6 ## 64^@(5/6)@ $$
And ## = ((64)^@(1/6)@)^5 $$
Q2.7 ## = ((2^6)^@(1/6)@)^5 $$
Q2.8 ## = 2^5 $$

The multiplication sign used in all expressions is cross multiplication with the code **. Some subjects typed one * which is for dot multiplication sign. Some subjects placed spaces surrounding multiplication sign ** and division sign //. This is not necessary but it does help to separate the two operands clearly and to avoid confusion regarding to the end of exponential value.

(coded with space) a^m ** a^n or a^m // a^n

Some subjects forgot to place begin math code ## and end math code $$ surround the expression. This type of mistakes appeared more when expressions are within a paragraph of text than a line of expression. It is very important not to forget to place begin and end math codes so that the expression will be extracted to the translation engine for translation. Otherwise the expression will be considered as text and will be translated incorrectly. It is possible for an expression to continue on to the next line (example Q2.6). In this case, it is necessary to mark begin and end math code for each line because the software analyzes one line at a time.

The most difficult issue in this question is when to use the baseline indicator. Should it be used when the expression could potentially be confusing or strictly follow the rule. That is to always use dot 5 or quote whenever the expression resumes at regular level following an exponential value or a subscript value. Is it necessary to put baseline indicator before multiplication sign? The expression of a to the power of m multiplies a to the power of n should be written as a^m”**a^n or a^m**a^n. When one looks at this expression with common sense, a^m would imply that the exponential value stops there instead of rasing the multiplication of a^n together with m as well. With that logic the baseline indicator would not be used. There were many other instances in this question (Q2.1 through Q2.8) where many subjects used baseline indicator and many other subjects did not. So the use of baseline indicators was considered questionable situations and points were not taken out. Overall performance was 96.94 percent.

Q3. This question contains 7 instances that needed coding. Q3.1 is part of the introductory question and the rest (Q3.2 through Q3.7) is in the method. This questions involves basic expression, set reference and a special symbol “infinity” where the subjects had to look up from the table.

Q3.1 ## x^2 < 1 $$
Q3.2 ## x^2 < 1 $$
Q3.3 ## x^2"-1 < 0 $$
Q3.4 ## (x+1)(x-1) < 0 $$
Q3.5 ## (x+1)(x-1) $$
Q3.6 ## (-..*, -1), (-1, 1) $$
Q3.7 ## (1, ..*) $$

The subjects had no problem looking up the table for infinity symbol ..* which would translate to dots 6, 1-2-3-4-5-6. Some subjects did not put space between coordinates (right after comma) which made it difficult to read. One subject incorrectly added multiplication sign in the expression that multiplication was implied. In Q3.5 the code was incorrectly written as ## (x+1)**(x-1) $$ where ** is not necessary. The task of the Braille transcriber is to convert print to Braille exactly without adding or deleting from the original text.

A few subjects did not put space surrounding less than sign and made it very confusing to read. Less than sign is dots 5, 1-3 which could be misread as baseline indicator dot 5 and letter k dots 1-3. The expression x^2<1 written without space can be misread as x squared, baseline indicator, k1. Therefore, it is very important to have spaces surrounding relational operators. This can be done by modifying the software to detect relational operators and insert spaces automatically before translation.

The issue of baseline indicator came up again in this question. This time it is certainly required. The expression x squared minus 1 less than 0 in Q3.3 ## x^2"-1 < 0 $$ requires baseline indicator " because the exponential value stops at 2. Without the baseline indicator, it is possible to consider 2-1 as the exponential value. Overall performance was 98.60 percent.

4.5 Attitude Test

Participants were asked regarding level of satisfaction of using this text-based math translation method. From the scale from 1 to 5 (poor, fair, acceptable, good, very good), overall satisfaction of text-based math input method was 2.6 and overall satisfaction of RS Braille software was 2.8. This is in the range of "acceptable".

Besides exercise results and satisfaction level, the researcher followed up by interviewing with five participants for their attitude on text-based math input method. Participants were asked about usefulness of the technique, how they may use this to assist in learning mathematics and suggestions for further improvement. Opinions varied between finding the technique useful and difficult to use, more appropriate for users who are not familiar with Braille, very accessible for the blind to write math expressions, and using the technique as math learning tool between sighted teachers and visually impaired students. Excerpts from the interviews are presented as follows:

Participant 1: *“I don’t know Braille so this method is good for me. However, it is new to me and I had a little bit of time this understanding and practicing this method. If I have time to work on method more I should be able to do a better job. It is an interesting method and should help us produce more math books in Thailand a great deal. This helps support learning in mathematics for visually impaired students."*

Participant 2: *“I am a Braille transcriber so I know Braille and know Nemeth code. It was difficult for me to try to remember extra set of code and I felt a bit uncomfortable using this method. It would be good for new people in the field and volunteers wanting to help with math book production. After that, Braille transcribers or blind proofreaders can check For accuracy again”.*

Participant 3: *“I am a visually impaired material production staff but I have to tutor blind students at the center also. This technique is very accessible for me because everything is text-based. I can type all math expressions on a regular editor by myself. I can use this technique to produce math sheets or math books in Braille for my blind students. The nature of this technique being text-based is useful for both blind and low vision persons”.*

Participant 4: *“I teach math at elementary level at a school for the blind of the organization. I find this technique useful as a math learning tool between teachers and blind students. Imagine a math teacher and a blind students sitting together at a computer, without knowledge of Braille, the teacher and student can interact and discuss a math problem and type math expressions on the computer and the information can be read by both parties. Blind students can type math homework independently on the computer and submit to the teacher. The technique would help speed up both teaching and learning process”.*

Participant 5: *"I know Braille but I am not an expert especially in math production. I did not like having to look for the codes. I think if the list of function is available as a menu in the software and after selecting desired function, the code would be placed in the editor, it would be much easier for me. Since number of knowledgeable math transcriber is very small, it is necessary to have some kind of math translation software that works with Thai text"*.

CHAPTER V
DISCUSSION

There are several ways for blind students to access math materials. For those that can read Braille, it is the preferred method. The students may begin with literary Braille by reading regular numbers with simple math symbols first. As math reading/writing requirements become more complex, Nemeth Math Braille Code is usually taught to the students. Besides Braille, word description of math expressions written in a word processor is also used. LaTex is one example that employs this method. With the trend of graphical user interface, the preferred choice for sighted persons is math symbol selection or equation editor in Microsoft Word. MathML tool are also available in several applications. However, these options are not fully accessible by the blind.

Duxbury Braille Translator is a print-to-braille application that uses math expressions produced in Scientific Notebook math editor and Equation Editor in Microsoft Word. Math expressions in both applications need to be produced by sighted persons. Dragon Naturally Speaking is a speech recognition software that allows voice input into the computer. However, this software only recognizes regular words and sentences and the user needs to speak in English. Metroplex Voice Computing Inc. provides math speech input option through MathPad and MathTalk. Voiced version of MathPad allows student to do basic addition, subtraction, multiplication, and division. It has the ability to check answers and read back. MathTalk allows the user to voice math expressions from pre-algebra, algebra, trigonometry, calculus, statistics, and graphs (Metroplex Computing, 2003). By using combination of these applications, it is possible to speak math expressions into the computer, listen to spoken math expressions and convert into Braille. However, the user must pay attention to the result of recognition in the math editor and may have to further edit parts that were recognized incorrectly. Ideal users of these applications are sighted persons who want to save some time typing math expressions by speaking into the computer instead. While further studies on effectiveness of producing math expressions via speech input or graphics-based input would give us more information on math input method, this research focused on text-based method of material production in Braille format.

When considering production of math materials for the blind, it is possible to explore different options ranging from speech input of math expressions with some kind of speech recognition application, graphics input via MathType or text-based input. Software developed in this research uses text-based approach. The text-based math input method can be used to train math teachers and material production staff effectively. Based on the result of average performance of 95.32%, participants could type many math expressions accurately. This is especially true for the participants with math background and math Braille transcription experience. Since math expressions had to be analyzed and retype in linear format, these two groups of participants used their knowledge to help during the exercise. Math teachers could accurately determine different components of a complete fraction and insert proper opening and closing fraction signs. Braille transcribers used their knowledge on linear style of writing math in Braille and substituted proper codes for this technique. Although the technique seemed to be awkward at first, participants not familiar with Braille found this technique potentially quite useful. The satisfaction level being in the range of “acceptable” shows that the participants tried to understand this new technique and were interested in using it further. It is recognized that various background of the participants contributed to varying results even though the same training was given prior to the exercise. This is in agreement with the study on Learning Technologies for Enhancing Student Understanding of Mathematics. The teachers in this study received the same instructions, materials, and training prior to the study, yet they each demonstrated unique teaching styles and methods of relating to their students. These differences in teaching behaviors were most likely influenced by the teachers’ past classroom experiences and their educational philosophy about how students learn mathematics (Drickey, 2006).

Even though the participants typed most of the symbols correctly, there is really no room for mistakes when it comes to producing quality math materials. In real working situation, further proofreading would have to be done in the Braille file. As supported in the study on Quality of Braille Instructional Materials Produced in Texas Public Schools, students who are learning to read cannot be expected to learn easily from materials that contain a substantial number of errors. Even if students are able to use context clues to decode what was transcribed, they may have difficulty comprehending what they are reading. Braille transcriptions containing multiple errors per page could have a negative impact on the ease and pace of reading (Herzberg & Stough, 2009).

Although math materials can be produced in Braille in many ways, technology especially Braille translation software has definitely impacted production process. According to the survey on production of textbooks and instructional materials in the United States, the trend was for states to rely on computer-translation software for Braille production, with direct input and manual brailing as backup systems. Although some steps have been taken toward improvement, most states were not using technological options for the delivery of specialized materials (Wall & Corn, 2002).

Since this method is text-based and the aim was to design short combination of keyboard characters to represent math symbols, it is not appropriate to compare this method with graphical solutions like MathType, MathML graphics tool or Scientific Notebook. However, it is recognized that sighted transcribers would feel more comfortable with graphics-based tools. Three other text-based candidates are LaTex, MathML codes and Megadots. Since both LaTex and MathML are based on longer word description, the closest method to this research is Megadots (Holiday, 1998). While some symbols may be similar, there are clear differences when considering the entire list of math symbol representations with Megadots. In order to implement various math symbols in Megadots, one needs to use characters from the keyboard as well as selecting options from the pull-down menu which is powerful and user-friendly in its own rights. Although one may be uncomfortable by having to type all symbols manually according to the method in this research, the list available is quite comprehensive and the advantage of this technique is that it can be used with any basic text editor like Notepad.

Due to the fact that the potential users of this technique (math teachers or material production staff) must be able to interpret the meaning of visual math expression properly and make decision to insert additional text-based symbol, it is necessary to give proper guidance to the potential users in order to have an effective implementation. This is especially true for the explanation of essential implied symbols such as exponential sign, base line indicator, open and closing fraction signs as well as distinguishing between various types of fractions.

It is understandable that the users may feel uncomfortable with this method at first. Three participants expressed, “It was difficult for me to try to remember extra set of code”, “It is new to me and I had a little bit of time understanding and practicing this method.” And “I did not like having to look for the codes.” This attitude was also found during the study on teachers’ perceptions of need for and Competency in Transcribing Braille materials in the Nemeth Code. Participants reported feeling much less competent to transcribe materials in higher-level math areas, particularly those skills requiring the transcription of graphics, than in basic elementary math areas. Teachers reported feeling more anxiety about transcribing math materials as the level of complexity increases (DeMario & Lian, 2000).

Even though the application of this research was mainly for production of math materials in Braille, the text-based method of typing math expressions can be used to apply toward math study as well. As a participant pointed out during the interview, “Imagine a math teacher and a blind students sitting together at a computer, without knowledge of Braille, the teacher and student can interact and discuss a math problem and type math expressions on the computer and the information can be read by both parties”. The nature of linear text-based style which is accessible when a blind student uses the computer with speech or Braille output, can act as an effective learning tool to bridge the gap between a sighted teacher and a blind student.

5.1 Implications

1. The most important part of this research was to see if math symbols could be produced via regular characters from the computer keyboard. Then this text-based math input method would be translated into Braille according to Nemeth math Braille code. The result of the research shows that it is possible to produce math in Braille

using this method. This is an alternative input technique as a contribution to math Braille production process in the field of educational services for the blind.

2. Given the fact that this method is fully accessible by visually impaired users, another potential benefit is to use this method as learning tool. A blind student may learn the method and write math expressions on a word processor to exchange with sighted teachers. Although both the student and the teacher may need to familiarize themselves with this input method, both parties should be able to understand what is written once initial learning curve has taken place.

5.2 Limitations

1. The number of 10 participants is considered a small sample size for the research. Each participant came from different educational centers of an organization with various level of expertise. While some were math teachers and experienced Braille transcribers, others were just newly hired and familiar with basic use of computers. While most participants were sighted, three participants were visually impaired. The text-based nature of this method which is fully accessible made it possible for visually impaired persons to participate in the exercise. In light of short evaluation period of these ten participants, the entire research project team considered software performance very important. Evaluation of the software began since development stage. The research team was heavily involved in testing the software. During the Thailand Annual Braille Contest in 2008, the first version of the software was distributed to material production staff and teachers of the visually impaired for evaluation and feedback. Valuable comments were gathered and the software was modified prior to the performance test with the participants. If there were more participants in this research and more time was given to the training as well as expanding the time to collect data by following up with everyone as they try to use this method in their work, more information would be gained regarding the performance and effectiveness of the method.

2. Although the software translated necessary math symbols used in the exercise, it needs to be tested more thoroughly for all symbols and combination of

various symbols written next to each other. As pointed out during the analysis of exercise section, some math symbols should be redesigned to avoid misunderstanding. Attempts should be made to present symbols as intuitive and logical as possible.

3. The software was developed and used during the training without having user manual. It is recognized that training is necessary for participants to understand this method before doing the exercise. However, live training is not always possible. For those that are interested in using the software in the future, it is necessary to have user manual with exercises available.

5.3 Recommendations for further studies

1. The software should be further developed by redesigning Print typing Code for problematic math symbols in order to avoid misunderstanding from user's perspective.

2. Conduct additional research on text-based input method with larger sample size among material production staff.

3. A study on application of the software as a learning tool should be conducted in order to bridge the gap between sighted math teachers and blind students.

4. User manual of the software should be developed and tested by letting participants study the manual prior to doing exercises.

5. The software should be tested in more details in terms of producing entire math textbook by using this text-based math input method.

6. Conduct additional research to explore other options for accessibility and production of math materials for visually impaired students such as speech feedback, speech recognition or word description of math symbols.

CHAPTER VI
CONCLUSION

Research on the development of a text-based math Braille translation software to enhance learning in mathematics was an attempt to provide material production staff and math teachers who are not Braille literate with the ability to produce Braille math materials in Nemeth Braille code standard. With advances in technology especially in Braille translation software where print text can be converted to Braille quickly and accurately, math Braille translation software is considered an additional important step toward improving material production process. Although many software developers have been contributing toward this challenge with a large emphasis on graphics-based solutions, the unique aspect of this research is its text-based approach where all math symbols can be input by typing regular characters from the computer keyboard. Although the nature of this text-based approach seemed awkward for sighted users who are more familiar with graphics approach at first, the performance of the software shows that the participants could adapt to text-based approach and produce quite accurate results. In addition, the text-based nature allowed preparation of documents with math expressions on any basic text editor and visually impaired persons especially blind computer users could fully utilize this technique.

The objectives of the research were (1) to develop text-based math Braille translation software module and interface it with a Thai Braille translation software and (2) to find out the performance of this software among math teachers and material production staff as well as their opinion in order to facilitate mathematics learning process of children with visual impairment.

6.1 Software development

A total of 192 math symbols was gathered from Nemeth Reference and other resources. The researcher designed print representation of these math symbols by using combination of printable characters on the keyboard without duplications. For example, one asterisk * is used for dot multiplication sign and two asterisks ** are used for cross multiplication sign. Text-based math expression of x^2”+1 would represent x squared follows by plus 1 where quotation mark is used to mark the end of exponential value or back to base line level. On the other hand x^n+1 would represent x to the power of n+1. In order to mix text and math expressions in the same document, begin math code ## and end math code $$ were used. After the complete list of math symbols with text-based design was compiled, the same table was used to create the search table before writing the source code. All 192 “Text-based print code” along with their corresponding “Braille Nemeth code” were rearranged according to groups from large to small. This arrangement was necessary for the source code to use by linear search.

The text-based math Braille translation engine was developed as a program using Pascal language. The program acted as a software module that could be called from the main software. Although this software module can be plugged into other Braille translation software of other languages, it was interfaced to a Thai Braille translation software called RS Braille. Text-based math string found by RS Braille would be passed to the engine for analysis. The engine would further loop through the string and linear search technique based on the table would be performed at each position of the string. If a match was found the Braille Nemeth code would be substituted. After the entire string was processed, the result would be sent back to the main software.

The main software had to be modified to interface with the engine. Whenever the software found a math string marked by begin math code ## and end math code $$, that portion would be extracted and the engine would be called. The result from the engine in US computer Braille code was then replaced at the same location of the document. Text and math expressions can be typed in the editor of RS Braille or copied from any text editor that was used to prepare source document with coded math symbols.

RS Braille software with text-based math Braille translation capability was heavily tested by software development team prior to testing it with participants in this research. The software was also tested during the Thailand Annual Braille Contest in 2008. An evaluation version of the software was distributed to material production staff and teachers of the visually impaired for evaluation and feedback. Valuable comments were gathered and the software was modified prior to the performance test with the participants.

6.2 Performance test and User opinion

The software was tested by 10 participants working as material production staff and math teachers from Christian Foundation for the Blind in Thailand. There were 7 sighted persons, 2 low vision persons and 1 blind person.

Initially the participants were given training on how to use the software and how to enter math expressions with this text-based method. During the three-hour training, the participants received introductory instruction of RS Braille software on how to enter text-based math symbols by using begin and end math indicators and focused on mathematical topics including basic operators, relational operators, simple fraction, mixed fraction, complex fraction, radical, indexed radical, exponential, subscript and baseline indicator. Even though the participants were not required to have knowledge of math Braille, the researcher emphasized the fact that everything had to be typed in linear style from left to right and special attention for proper opening and closing indicators which may not exist in print had to be given. The use of base line indicator was explained in detail with specific examples as the symbol determined when an exponential value or a subscript value was finished.

After the training, all participants worked on the exercises for 2 hours. They were given print exercises and the task was to type all 20 math expressions (N1 – N20) and 3 questions with text (Q1 - Q3) in RS Braille editor with proper text-based math symbols. One low vision person read print exercises independently. The researcher dictated the exercises to the other low vision person and the blind person.

Both print and translated Braille files of all participants were copied and printed out for analysis.

Total number of points for the entire exercise is 280 where the first set of 20 math expressions (N1 – N20) is 125 and questions with text (Q1 - Q3) have total points of 20, 85 and 50 respectively. Out of 280 possible points, all participants earned the average of 266.90 points which is equal to 95.32 percent accuracy.

Participants were asked regarding level of satisfaction of using this text-based math translation method. From the scale from 1 to 5 (poor, fair, acceptable, good, very good), overall satisfaction of text-based math input method was 2.6 and overall satisfaction of RS Braille software was 2.8. This is in the range of “acceptable”.

Besides exercise results and satisfaction level, the researcher followed up by interviewing with five participants for their attitude on text-based math input method. Participants were asked about usefulness of the technique, how they may use this to assist in learning mathematics and suggestions for further improvement. Opinions varied between finding the technique useful and difficult to use, more appropriate for users who are not familiar with Braille, very accessible for the blind to write math expressions, and using the technique as math learning tool between sighted teachers and visually impaired students.

6.3 Conclusion

It is clear that this text-based math symbol input method does work and can produce quite accurate Braille for mathematical expressions. In addition to application toward Braille production, this technique can potentially be applied as a learning tool between sighted math teacher and blind students as well. Math teachers and material production staff do not need to remember actual Braille code. The technique can be used by both sighted and visually impaired persons. Even though it may be difficult to use at first, once math teachers and material production staff are familiar with the method, learning process in mathematics of visually impaired students should be enhanced a great deal. In addition, text files of coded math text books can be produced before translation by a larger group of volunteers with any text editor without having to install special software. Furthermore, the engine can be applied to Braille translation software of other languages besides Thai. As we strive to provide equal educational opportunity for all children in the world, text-based approach of this research should contribute toward improving learning environment for blind students especially those that are living in developing countries.

REFERENCES

Acharya (2009). *Nemeth Braille code for mathematical and scientific notations*. http://acharya.iitm.ac.in/disabilities/nemeth_brl.php

American Printing House for the Blind (1972). *The Nemeth Braille Code for Mathematics and Science Notation 1972 Revision (Nemeth resource).* www.aph.org

Basic Mathematics for 10th grade - Book one (2005). *Three instances of text with math formulas Q1 – Q3*. The Institute for the Promotion of Teaching Science and Technology, Thailand Ministry of Education, 2, 90, 104.

Bitter, G. G., & Hatfield, M. M. (1998). The role of technology in the middle grades. In L. Leutzinger (Ed.), *Mathematics in the middle*, 36-41. Reston, VA : National Council of Teachers of Mathematics.

Bogart, D. (2009). Louis Braille Celebration: Unifying the English Braille Codes. *Journal of Visual Impairment & Blindness, 103*(10), 581-583.

Corn, A. L., & Koenig, A. J. (2002). Literacy for students with low vision: A framework for delivering instruction. *Journal of Visual Impairment & Blindness, 97*, 305-321.

D'Andrea, F. M., & Farrenkopf, C. (Eds.). (2000). *Looking to learn: Promoting literacy for students with low vision*. New York: AFB Press.

DeMario, N. C., & Lian, M. J., (2000). Teachers' Perceptions of Need for and Competency in Transcribing Braille Materials in the Nemeth Code. *Journal of Visual Impairment & Blindness, 94*, 7-14.

Dot Less Braille (2009). *Linear method of expressing mathematics by Nemeth Braille code*. http://www.dotlessbraille.org/NemethIntro.htm)

Drickey, N., (2006). Learning Technologies for Enhancing Student Understanding of Mathematics. *International Journal of Learning, 13*(5), 109-116.

Farrenkopf, C. (2008). Reading Instruction: Best Practices and Realities in Canada's Largest School District. *Journal of Visual Impairment & Blindness, 102*(4), 200-203.

Grey, C., (2009). Louis Braille Celebration: Lessons from the Development of Computer Braille Code. *Journal of Visual Impairment & Blindness, 103,* 740-741.

Herzberg, T. S., & Stough, L. M. (2009). The Quality of Brailled Instructional Materials Produced in Texas Public Schools. *Journal of Visual Impairment & Blindness, 103,* 722-733.

Holiday, D. (1998). *Megadots Braille Translator. Raised Dots Computing, and Duxbury Systems* Inc.http://www.duxburysystems.com/ documentation/ megadots/chap22.htm.

Inaram, K. (2004). *Learning Braille (Nemeth resource).* Special Education Program Ratchapat Suandusit University.

Koenig, A. J., & Holbrook, M. C. (2000). Ensuring high-quality instruction for students in braille literacy programs. *Journal of Visual Impairment & Blindness, 94,* 677-694.

Lofstedt, B. (2006). *WinBraille Translator, Windows Equation Editor math support. Index Braille*. http://www.indexbraille.com/Support/Knowledge-base/WinBraille-5/Emboss-mathematics.aspx.

Mani, M. (2005). *Mathematics Made Easy for Children with Visual Impairment (Nemeth resource).* International Council for Education of People with Visual Impairment.

Mani, M. (2006). WIMAT (Webel - *ICEVI Mathermatics Transcription Software).* Workshop presentation at 2006 ICEVI World Conference.

McKenzie, M. (2009). *SCSDB aces Braille challenge. LEE G HEALY*. Spartanburg Herald - Journal. Spartanburg, S.C.: Feb 6, 2009.

Metroplex Computing (2003). *MathPad and MathTalk product description. Metroplex Computing* Inc. http://metroplexvoice.com/products.htm

National Braille Press (2009). *Louis Braille's life*. National Braille Press Inc. http://www.nbp.org.

National Federation of the Blind (2004). *The National Center for Blind Youth in Science (NCBYS)*. The National Federation of the Blind Jernigan Institute. http://www.nfb.org.

National Federation of the Blind (2009). *Few blind people in the US are using Braille.LEE G HEALY*. Spartanburg Herald - Journal. Spartanburg, S.C.: Mar 30, 2009.

Niyomphol, W. (1989). *Research project on Thai Braille Translation software for DOS (TBT)*.frwny@mahidol.ac.th.

Niyomphol, W., and Nantachaipitak, W. (2002). *Research project on Thai Braille Translation software for Windows (TBTW)*. Ratchasuda College, Mahidol University. www.rs.mahidol.ac.th.

Niyomphol, W., Tandayya P., and Nantachaipitak, W. (2008). *Research project on Ratchasuda Braille (RS Braille)*. Ratchasuda College, Mahidol University; and Engineering Department, Prince of Songkla University. www.rs.mahidol.ac.th and www.psu.ac.th

Niyomphol, W. (2010). *Report on math/science promotion for the blind in Thailand*. Ratchasuda College, Mahidol University.

Ontario Ministry of Education. (2005). *Education for all: The report of the expert panel on literacy and numeracy instruction for students with special education needs, kindergarten to grade 6*. Toronto: Ontario Ministry of Education.

Sullivan, J. (1997). *Duxbury Braille Translator, math built-in menu system*. Duxbury systems Inc. www.duxburysystems.com.

Sullivan, J. (2001). *Duxbury Braille Translator, Scientific Notebook and math LaTex importation*. Duxbury systems Inc. www.duxburysystems.com.

Sullivan, J. (2008). *Duxbury Braille Translator, Mathtype Microsoft Word importation*. Duxbury systems Inc. www.duxburysystems.com.

Sullivan, J. (2009). *Unifying Braille: Louis Braille's Vision in Tomorrow's World*. Paper presentation at the Celebration of the Bicentenary of Louis Braille's birth at UNESCO Headquarter, Paris.

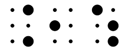
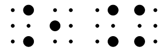
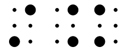
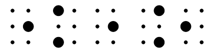
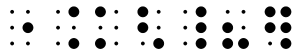
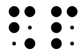
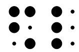
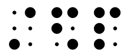
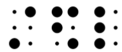
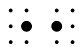
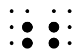
Wall, R. S., & Corn, A. L., (2002). Production of Textbooks and Instructional Materials in the United States. *Journal of Visual Impairment & Blindness, 96*, 212-222.

APPENDIX

Appendix A: Math Table with Braille representations
Numeric key, Picture, English description, Thai description, Print typing code, Braille representation
Mahidol University

ที่	รูปภาพ	อังกฤษ	ไทย	รหัสพิมพ์ อักษรปกติ	คอมพิวเตอร์เบรลล์
001	+	plus operator	เครื่องหมายบวก	+	⠬
002	−	minus operator	เครื่องหมายลบ	-	⠤
003	×	multiplication ×	เครื่องหมายคูณ เอ็กซ์	**	⠈⠡
004	•	multiplication dot	เครื่องหมายคูณ จุด	*	⠡
005	÷	division	เครื่องหมายหาร	//	⠠⠌
006	⟌	division linear	เครื่องหมายหารยาว	-/	⠕
007	⟌	division spatial	เครื่องหมายหาร ว่าง	-/-	⠒⠒⠒ ⠕
008	/	division slash	ทับ	/	⠸⠌
009	±	plus or minus	เครื่องหมายบวกหรือลบ	+|-	⠬⠤
010	∓	minus or plus	เครื่องหมายลบหรือบวก	-|+	⠤⠬
011	+−	plus followed by minus	เครื่องหมายบวกและลบ	+&-	⠬⠐⠤
012	−+	minus followed by plus	เครื่องหมายลบและบวก	-&+	⠤⠐⠬

013	$--$	minus followed by minus	เครื่องหมายลบและลบ	-&-	
014	$\cap$	intersection	อินเตอร์เซ็กชั่น	*:	
015	$\cup$	union	ยูเนียน	*+	
016	$\sim$	tilde	นิเสธ	~	
017	$\vee$	logical sum	โลจิคอล ผลรวม	@ls	
018	$\wedge$	logical product	โลจิคอล ผลคูณ	@lp	
019	&	ampersand	เครื่องหมายแอ่น	&	
020	$\vert$	vertical bar	โดยที่	|	
021	$\circ$	hollow dot	จุดกรวง	.○	
022	$\backslash$	backslash	เครื่องหมายแบคแสลท	\	
023	$=$	equals	เครื่องหมายเท่ากับ	=	
024	$<$	less than	เครื่องหมายน้อยกว่า	<	
025	$>$	greater than	เครื่องหมายมากกว่า	>	
026	$\leq$	less than with bar under	เครื่องหมายน้อยกว่าหรือเท่ากับ	<-	
027	$\leqq$	less than with equal sign under	เครื่องหมายน้อยกว่าหรือเท่ากับ (ด้านล่าง)	<=	

028	$\geq$	greater than with bar under	เครื่องหมายมากกว่าหรือเท่ากับ	>-	
029	$\geqq$	greater than with equal sign under	เครื่องหมายมากกว่าหรือเท่ากับ (ด้านล่าง)	>=	
030	$\neq$	unequal	เครื่องหมายไม่เท่ากับ	/=	
031	$<>$	less than followed by greater than	เครื่องหมายน้อยกว่าและมากกว่า	<>	
032	$><$	greater than followed by less than	เครื่องหมายมากกว่าและน้อยกว่า	><	
033	$\stackrel{?}{=}$	equal sign with question mark over	เครื่องหมายเท่ากับและเครื่องหมายคำถาม	?=	
034	$\perp$	perpendicular to	ตั้งฉาก	|_	
035	$\parallel$	parallel to	ขนาน	|=	
036	$\not\perp$	not perpendicular to	ไม่ตั้งฉาก	/|_	
037	$\nparallel$	not parallel to	ไม่ขนาน	|/=	
038	$:$	ratio	อัตราส่วน	:	
039	$::$	proportion	สัดส่วน	::	

040	≈	double tilde	โดยประมาณ	~~	⠈⠱⠈⠱
041	≅	tilde with equal sign under	เท่ากันทุกประการ	~=	⠈⠱⠨⠅
042	⌒	arc, concave upward	เส้นโค้งคว่ำ	!^	⠫⠁⠠
043	‿	arc, concave downward	เส้นโค้งหงาย	!v	⠫⠄
044	∈	is an element of	เป็นสมาชิกของ	&e	⠈⠑
045	∋	reverse membership	สมาชิกผกผัน	/&e	⠈⠢
046	≍	equivalence	จำนวนเท่ากัน	==	⠈⠣⠠⠣
047	≡	identity	สมนัยกัน	&id	⠸⠳
048	∝	variation	แปรผัน	=*	⠸⠿
049	⊂	inclusion	เป็นสับเซตของ	=&	⠸⠐⠅
050	⊃	reverse inclusion	ไม่เป็นสับเซตของ	/=&	⠸⠨⠂
051	⊆	inclusion with bar under	เครื่องหมายประกอบด้วย	=&-	⠸⠐⠅⠱
052	⫅	inclusion with equal sign under	เครื่องหมายประกอบด้วย, เท่ากับด้านล่าง	=&=	⠸⠐⠅⠨⠅
053	⊇	reverse inclusion with bar under	เครื่องหมายประกอบด้วย, เส้นด้านกลับ	=&/-	⠸⠨⠂⠱

054	$\supseteqq$	reverse inclusion with equal sign under	เครื่องหมายเส้นด้านกลับ, เท่ากับด้านล่าง	=&/=	⠸⠨⠂⠨⠅
055	$\underline{\cap}$	intersection with bar under	อินเตอร์เซ็กชั่น มีเส้นด้านล่าง	*:-	⠨⠩⠱
056	$\underline{\cup}$	union with bar under	ยูเนียน มีเส้นด่านล่าง	*+-	⠨⠬⠱
057	$\rightarrow$	contracted form of right-pointing arrow	ลูกศรชี้ขวา ขนาดเล็ก	->	⠫⠕
058	$\longrightarrow$	ordinary shaft, right–pointing arrow	ลูกศรชี้ขวาปกติ	-h>	⠫⠒⠒⠕
059	$\Longrightarrow$	double shaft, right–pointing arrow	ลูกศรชี้ขวาสองเท่า	-hh>	⠫⠶⠶⠕
060	$\leftarrow$	ordinary shaft, left-pointing arrow	ลูกศรชี้ซ้ายปกติ	-h<	⠫⠪⠒⠒
061	$\Leftarrow$	double shaft, left -pointing arrow	ลูกศรชี้ซ้ายสองเท่า	-hh<	⠫⠪⠶⠶
062	$\uparrow$	arrow pointing up	ลูกศรชี้ขึ้น	|^	⠫⠣⠒⠒⠕
063	$\downarrow$	arrow pointing down	ลูกศรชี้ลง	|v	⠫⠩⠒⠒⠕
064	$\longleftrightarrow$	ordinary shaft, double-pointing arrow	ลูกศรซ้ายขวา	-h<>	⠫⠪⠒⠒⠕
065	$\Longleftrightarrow$	double shaft, double-pointing arrow	ลูกศรซ้ายขวาสองเท่า	-hh<>	⠫⠪⠶⠶⠴

066	⇄	right-pointing over left-pointing arrow	ลูกศรชี้ขวา อยู่เหนือลูกศรชี้ซ้าย	-><	⠫⠒⠒⠕⠫⠪⠒⠒
067	⇆	left-pointing over right-pointing arrow	ลูกศรชี้ซ้าย อยู่เหนือลูกศรชี้ขวา	-<>	⠫⠪⠒⠒⠫⠒⠒⠕
068	↓↑	down-pointing followed by up-pointing arrow	ลูกศรชี้ลง ลูกศรชี้ขึ้น	-v^	⠫⠩⠒⠒⠕⠐⠫⠣⠒⠒⠴
069	↑↓	up-pointing followed by down-pointing arrow	ลูกศรชี้ขึ้น ลูกศรชี้ลง	^v-	⠫⠣⠒⠒⠕⠐⠫⠩⠒⠒⠕
070	↕	arrow pointing up and down	ลูกศรขึ้น และลง	-^v	⠫⠣⠪⠒⠒⠕
071	(	parenthesis, left	เครื่องหมายวงเล็บเปิด	(	⠷
072	)	parenthesis, right	เครื่องหมายวงเล็บปิด	)	⠾
073	[	bracket, left	วงเล็บก้ามปูเปิด	[	⠈⠷
074	]	bracket, right	วงเล็บก้ามปูปิด	]	⠈⠾
075	{	brace, left	วงเล็บปีกกาเปิด	{	⠨⠷
076	}	brace, right	วงเล็บปีกกาปิด	}	⠨⠾
077	〈	angle bracket, left	วงเล็บมุมเปิด	&[	⠨⠨⠷

078	〉	angle bracket, right	วงเล็บมุมปิด	&]	
079	||	vertical bar, double	เส้นตั้งตรงสองเส้น	||	
080	(	enlarged parenthesis, left	วงเล็บใหญ่เปิด	*(	
081	)	enlarged parenthesis, right	วงเล็บใหญ่ปิด	)*	
082	[	enlarged bracket, left	วงเล็บก้ามปูใหญ่เปิด	'[	
083	]	enlarged bracket, right	วงเล็บก้ามปูใหญ่ปิด	]'	
084	{	enlarged brace, left	วงเล็บปีกกาใหญ่เปิด	‘{	
085	}	enlarged brace, right	วงเล็บปีกกาใหญ่ปิด	}’	
086	|	enlarged vertical bar, left and right	เส้นตรงหนาซ้ายและขวา	*|	
087	{ }	empty set	เซ็ตว่าง	{}	
088		transcriber's grouping symbol, opening & closing	สัญลักษณ์เปิดปิดกลุ่ม	"	
089		simple fraction indicator, opening	เครื่องหมายเปิดเศษส่วน	@(	
090		simple fraction indicator, closing	เครื่องหมายปิดเศษส่วน	)@	

091	———	simple fraction line, horizontal	เส้นแสดงเศษส่วน	/	⠌
092	/	simple fraction line, diagonal	เส้นแสดงเศษส่วนเอียง	|/	⠸⠌
093		fractional part of mixed number indicator, opening	เปิดเศษส่วนคระ	@[	⠸⠹
094		fractional part of mixed number indicator, closing	ปิดเศษส่วนคระ	]@	⠸⠼
095		complex fraction indicator, opening	เปิดเศษส่วนซ้อน	@{	⠠⠹
096		complex fraction indicator, closing	ปิดเศษส่วนซ้อน	}@	⠠⠼
097	———	complex fraction line, horizontal	เส้นแสดงเศษส่วนซ้อน	@/	⠠⠌
098	/	complex fraction line, diagonal	เส้นแสดงเศษส่วนซ้อนเอียง	@|/	⠠⠸⠌
099		shape indicator	เครื่องหมายแสดงรูปทรง ปกติ	?r	⠫
100		Shaded shape indicator	เครื่องหมายแสดงรูปทรง หนา	?b	⠨
101		filled- in shape indicator	เครื่องหมายแสดงรูปทรง ทึบ	?f	⠸
102		circle	รูปวงกลม	!c	⠫⠉
103		triangle	รูปสามเหลี่ยม	!t	⠫⠞
104		right triangle	รูปสามเหลี่ยมมุมฉาก	!tr	⠫⠞⠨⠗⠻

105	□	square	รูปสี่เหลี่ยมจัตุรัส	!s	⠫⠲
106	▭	rectangle	รูปสี่เหลี่ยมผืนผ้า	!r	⠫⠗
107	◇	diamond	รูปสี่เหลี่ยมขนมเปียกปูน	!d	⠫⠙
108	⬠	pentagon	รูปห้าเหลี่ยม	!p	⠫⠢
109	⬡	hexagon	รูปหกเหลี่ยม	!h	⠫⠖
110	☆	star	รูปดาว	?*	⠫⠎
111	⬭	ellipse	รูปวงรี	!e	⠫⠑
112	∠	angle	มุม	![	⠫⠪
113	∟	right angle	มุมฉาก	![r	⠫⠪⠨⠗⠻
114	—	bar, horizontal, modifying	เส้นแนวนอน	!-	⠱
115	⊕	circle with interior plus sign	เครื่องหมายบวกภายในวงกลม	!c+	⠫⠉⠸⠫⠬⠻
116	⊙	circle with interior dot	จุดภายในวงกลม	!c.	⠫⠉⠸⠫⠡⠻

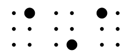
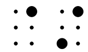
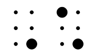
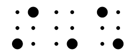
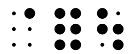
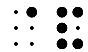
117	α	alpha	แอลฟา	'a	⠨⠁
118	β	beta	เบตา	'b	⠨⠃
119	γ	gamma	แกมมา	'g	⠨⠛
120	δ	delta	เดลตา	'd	⠨⠙
121	ε	epsilon	เอปซิลอน	'e	⠨⠑
122	ζ	zeta	เซตตา	'z	⠨⠵
123	η	eta	เอตา	'et	⠨⠱
124	θ	theta	ธีตา	'th	⠨⠹
125	ι	iota	ไอโอตา	'I	⠨⠊
126	κ	kappa	แคปปา	'k	⠨⠅
127	λ	lambda	แลมดา	'l	⠨⠇
128	μ	mu	มิว	'm	⠨⠍
129	ν	nu	นิว	'n	⠨⠝
130	ξ	xi	ซาย	'x	⠨⠭

131	ο	omicron	โอไมครอน	'o	⠨⠕
132	π	pi	พาย	'p	⠨⠏
133	ρ	rho	โร	'r	⠨⠗
134	σ	sigma	ซิกมา	's	⠨⠎
135	τ	tau	ทาว	't	⠨⠞
136	υ	upsilon	อุปซิลอน	'u	⠨⠥
137	φ	phi	ฟาย	'ph	⠨⠋
138	χ	chi	คาย	'ch	⠨⠯
139	ψ	psi	พซาย	'ps	⠽
140	ω	omega	โอเมกา	'om	⠨⠺
141	ϡ	sampi	แซมพาย	'c	⠨⠉
142		numeric indicator	เครื่องหมายนำเลข	?#	⠼
143		letter sign	เครื่องหมายนำตัวอักษร	?;	⠰
144		punctuation indicator	เครื่องหมายนำสัญลักษณ์	?_	⠤

145		superscript indicator	เครื่องหมายยกกำลัง	^	⠘
146		subscript indicator	เครื่องหมายห้อย	;	⠰
147		comma or comma-space within subscript or superscript	คอมม่าภายในเครื่องหมายยกกำลังหรือห้อย	—,	⠪
148		baseline indicator	เครื่องหมายนำแถวปกติ	"	⠐
149		cancellation indicator, opening	เครื่องหมายเปิดยกเลิก	@#	⠪
150		cancellation indicator, closing	เครื่องหมายปิดยกเลิก	#@	⠻
151		carried-number indicator for addition (varying in length)	เครื่องหมายทดกันบวก	·—·	⠶⠶⠶⠶⠶
152		directly-over indicator	เครื่องหมายแสดงการอยู่ด้านบน	?^	⠣
153		directly-under indicator	เครื่องหมายแสดงการอยู่ด้านล่าง	?v	⠩
154		termination indicator	เครื่องหมายจบสัญลักษณ์	?]	⠻
155		general reference indicator	เครื่องหมายอ้างอิงทั่วไป	?-?	⠈ ⠻
156		italic indicator	เครื่องหมายแสดงตัวเอียง	?:	⠨
157		italic type-form indicator, opening	เครื่องหมายแสดงตัวเอียงเปิด	:(	⠠ ⠄ ⠨

158		italic type-form indicator, closing	เครื่องหมายแสดงตัวเอียงปิด	):	⠨⠠⠄
159		bold type-form indicator, opening	เครื่องหมายแสดงตัวหนาเปิด	_(	⠠⠄⠸
160		bold type-form indicator, closing	เครื่องหมายแสดงตัวหนาปิด	)_	⠸⠠⠄
161	%	percent sign	เครื่องหมายเปอร์เซ็นต์	%	⠈⠴
162	$	dollar sign	เครื่องหมายดอลลาร์	$	⠈⠎
163	¢	cent sign	เซ็น	?c	⠈⠉
164	£	pound sterling	เครื่องหมายปอนด์	?l	⠈⠇
165	°	degree sign	เครื่องหมายองศา	.dg	⠘⠨⠡
166	?	omission sign, general	เครื่องหมายยกเว้น	?no	⠿
167	√	radical	เครื่องหมายราก	^[	⠜
168	$\sqrt{\sqrt{\ }}$	first inner radical indicator	เครื่องหมายรากระดับที่หนึ่ง	^*	⠨
169	$\sqrt{\sqrt{\sqrt{\ }}}$	second inner radical indicator	เครื่องหมายรากระดับที่สอง	^**	⠨⠨
170		index of radical	เครื่องหมายระบุรากที่	^#	⠣
171	*	asterisk	เครื่องหมายดอกจัน	?*	⠈⠼

172	#	crosshatch, pounds (weight)	เครื่องหมายช๊าบ	?#	⠨⠼
173	∧	caret	แคลอทชี้ขึ้น	_^	⠸⠣
174	∨	inverted caret	แคลอทชี้ลง	_v	⠸⠩
175	′	prime	ไพรม์	?'	⠠
176	|	tally	เครื่องหมายขีดนับ	#_	⠤
177	✓	check mark	เครื่องหมายถูก	'/	⠈⠜
178	〃	ditto mark	เครื่องหมายซ้ำ	'.	⠠⠄
179	∞	infinity	อนันต์ (ไม่มีที่สิ้นสุด)	..*	⠠⠿
180	@	at-sign	แอ๊ด	@	⠈⠁
181	∅	empty or null set	เซ็ตว่าง	{0}	⠸⠴
182	!	factorial	แฟคทอร์เรียล	&	⠯
183	∂	partial derivative	พาล์เทลดิริเวอทีป	@d	⠈⠙
184	∫	integral sign	อินทิเกริล	@t	⠞
185	∫∫	double integral sign	อินทิเกริล คู่	@@t	⠞⠞
186	∮	integral sign, with superposed circle	อินทิเกริล กับวงกลม	@tc	⠞⠈⠫⠉⠻

187	Å	angstrom unit	หน่วยวัดแองสตรอม	@as	
188	$\because$	since (because)	เพราะว่า	@v	
189	$\therefore$	therefore	เพราะฉะนั้น	@/	
190	$\not\therefore$	it does not follow that	ไม่เป็นไปตามนั้น	/@^	
191	$\exists!$	existential quantifier (there exists exactly one)	จำนวน, ขนาด	@eq	
192	$\forall$	universal quantifier	ยูนิเวอร์ซอล คอทิไฟเออร์	@uq	

BIOGRAPHY

NAME	Mr. Wiraman Niyomphol
DATE OF BIRTH	26 January 1964
PLACE OF BIRTH	Kon Kaen, Thailand
INSTITUTIONS ATTENDED	West Chester University, 1983-1987 Bachelor of Science in Computer Science Mississippi State University, 1989-1991 Master of Education in Computer Access Technology for the Visually Impaired Mahidol University, 2005-2010 Doctor of Philosophy in Science and Technology Education
HOME ADDRESS	95/8 Mu 6 Jatupitaporn Gratumlom, Sampran, Nakornpatom 73220 Thailand