

## ภาคผนวก A

# การทดลองที่ 1 ข้อมูลและคณิตศาสตร์ในคอมพิวเตอร์

การทดลองนี้เป็นการทบทวนความเข้าใจและแบบฝึกหัดเสริมของเนื้อหาในบทที่ 2 เนื่องจากจำนวนบิตข้อมูลที่ยาวขึ้นจำเป็นต้องใช้โปรแกรมคอมพิวเตอร์ช่วยคำนวณแทน โดยมีวัตถุประสงค์ ดังต่อไปนี้

- เพื่อให้เข้าใจการแปลงและคณิตศาสตร์สำหรับเลขจำนวนเต็มฐานสองชนิดไม่มีเครื่องหมายและมีเครื่องหมายแบบ 2-Complement
- เพื่อให้เข้าใจการแปลงและคณิตศาสตร์สำหรับเลขทศนิยมฐานสองมาตรฐาน IEEE754 ชนิด Single Precision
- เพื่อให้เข้าใจรหัส ASCII และ Unicode สำหรับข้อมูลตัวอักษร

นอกจากเนื้อหาในบทที่ 2 แล้ว ผู้อ่านสามารถศึกษาเว็บเพจเพิ่มเติม เพื่อทำความเข้าใจอย่างลึกซึ้ง ได้แก่

- [https://www.tutorialspoint.com/cprogramming/c\\_data\\_types.htm](https://www.tutorialspoint.com/cprogramming/c_data_types.htm)
- <https://www3.ntu.edu.sg/home/ehchua/programming/java/daterepresentation.html>

ผู้อ่านจะพบว่าเนื้อหาในเว็บของมหาวิทยาลัยนันทยาง ประเทศสิงคโปร์ เป็นการสอนพื้นฐานภาษา Java ใช้งานข้อมูลเป็นเลขฐานสองเหมือนกับภาษา C/C++ ในเว็บที่สอง การทดลองจะครอบคลุมเนื้อหาตามทฤษฎี โดยจะเริ่มจากเลขจำนวนเต็ม เลขทศนิยม และตัวอักษรตามลำดับ

## A.1 การแปลงและคณิตศาสตร์สำหรับเลขฐานสองจำนวนเต็ม

### A.1.1 การทดลอง

เนื่องจากการแปลงเลขฐานสิบเป็นฐานสองชนิดไม่มีเครื่องหมาย (unsigned) ผู้อ่านสามารถใช้เครื่องคิดเลขทางวิทยาศาสตร์ทั่วไป ดังนั้น การทดลองนี้จะเน้นที่การแปลงเป็นเลขฐานสองชนิดมีเครื่องหมายแบบ 2 Complement สอดคล้องกับเนื้อหาในหัวข้อที่ 2.2 โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านกด กรอกหรือคลิกที่ ชื่อลิงค์ต่อไปนี [http://www.free-test-online.com/binary/signed\\_converter.html](http://www.free-test-online.com/binary/signed_converter.html) ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้

1. คลิกเลือกที่ปุ่ม Signed แล้วจึงกรอกเลข -123 ลงในกล่องข้อความ ดังรูปที่ A.1

The image shows a web-based signed binary converter interface. At the top, there is a text input field containing the number "-123". To the right of the input field is a label "Binary". Below the input field are two radio buttons: "Unsigned" and "Signed". The "Signed" radio button is selected. Underneath the radio buttons is a grid of buttons for various operations. The first row contains "Bin2Dec", "Dec2Hex", "Hex2Bin", and "CL". The second row contains "Dec2Bin", "Hex2Dec", "Bin2Hex", and "RoR". The third row contains "D", "E", "F", and "RoL". The fourth row contains "A", "B", "C", and "ShR". The fifth row contains "7", "8", "9", and "ShL". The sixth row contains "4", "5", "6". The seventh row contains "1", "2", "3". The eighth row contains "0", "2's C", and "+/-".

รูปที่ A.1: กรอกเลข -123 ลงในกล่องข้อความ และคลิกเลือกที่ปุ่ม Signed เพื่อให้โปรแกรมแปลงเลขจำนวนเต็ม -123 เป็นเลขฐานสองชนิด Signed ชนิด 2 Complement

หน้าต่างมีลักษณะคล้ายเครื่องคิดเลข ประกอบด้วยปุ่มต่างๆ ดังนี้

- 'Bin2Dec' 'Dec2Bin' สำหรับแปลงเลขฐานสองเป็นฐานสิบไปและกลับ
- 'Dec2Hex' 'Hex2Dec' สำหรับแปลงเลขฐานสิบเป็นฐานสิบหกไปและกลับ
- 'Hex2Bin' 'Bin2Hex' สำหรับแปลงเลขฐานสองเป็นฐานสิบหกไปและกลับ
- ปุ่ม 0-9 และ A-F สำหรับกรอกตัวเลขฐานสิบและฐานสิบหก
- CL (Clear) สำหรับล้างค่าในกล่องข้อความให้เป็น 0
- RoR (Rotate Right) และ RoL (Rotate Left) สำหรับเลื่อนวนเลขที่อยู่ในกล่องข้อความทางขวาและซ้าย ตามลำดับ
- ShR (Shift Right) และ ShL (Shift Left) โดยป้อนเลข 0 เข้ามาแทน
- 2's C (omplement) สำหรับแปลงเลขฐานสองให้เป็นค่า 2's Complement

- +/- สำหรับกลับเครื่องหมายของตัวเลขฐานสิบในกล่องข้อความ

2. กดปุ่มเครื่องหมาย 'Dec2Bin' เพื่อให้เป็นเลขฐานสองชนิด Signed ดังรูปที่ A.2

รูปที่ A.2: ผลลัพธ์จากการแปลงเลข -123 ให้เป็นเลขฐานสองชนิด Signed 2-Complement ความยาว 24 บิต

3. กดปุ่มเครื่องหมาย 'Bin2Hex' เพื่อแปลงเลขฐานสองที่ได้ให้เป็นเลขฐานสิบหกชนิด Signed ตามรูปที่ A.3

รูปที่ A.3: ผลลัพธ์จากการแปลงเลข -123 ให้เป็นเลขฐานสิบหก 6 หลักจากเลขฐานสองชนิด Signed 2-Complement ความยาว 24 บิต

4. กดปุ่ม Hex2Bin เพื่อแปลงผลลัพธ์เลขฐานสิบหกที่ได้กลับไปเป็นฐานสอง แล้วเลือกตัวเลขฐานสองทั้งหมด แล้วทำการคัดลอก (Copy) หรือกดปุ่ม Ctrl-C พร้อมกัน

5. คลิกบนชื่อลิงค์ต่อไปนี้ เพื่อเปิดหน้าเว็บสำหรับ บวก/ลบ/คูณ/หาร เลขจำนวนเต็ม ทั้งชนิด Unsigned และ Signed ต่อไปนี้

[http://www.free-test-online.com/binary/binary\\_calculator.html](http://www.free-test-online.com/binary/binary_calculator.html)

6. กดเลือกปุ่มออฟชั่น Signed ก่อนแล้วจึงทำการวาง (Paste) ลงในกล่องข้อความ เพื่อเปลี่ยนการทำงานให้อยู่ในโหมดตัวเลขฐานสองชนิดมีเครื่องหมายตามรูปที่ A.4

รูปที่ A.4: หน้าต่างวางเลขการแปลงเลข -123 ให้เป็นเลขฐานสองชนิด Signed 2-Complement ความยาว 24 บิต

7. กดปุ่ม '-' เพื่อทำการกระบวนการลบเลข แล้ววาง (Paste) เลข -123 อีกรอบในกล่องข้อความที่ว่างลง
8. กดปุ่ม = เพื่อแสดงผลลัพธ์

รูปที่ A.5: ผลลัพธ์เลขการแปลงเลข (-123) - (-123) ให้เป็นเลขฐานสองชนิด Signed 2-Complement ความยาว 24 บิต

ในรูปที่ A.5 แสดงให้เห็นว่า  $-123 - (-123) = 0$

## A.1.2 กิจกรรมท้ายการทดลอง

จงทำการทดลองและตอบคำถามต่อไปนี้ โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.2.2 และตรวจคำตอบตามวิธีทำการทดลองที่ได้ทำไป

1. จงแปลงเลขฐานสิบชนิดไม่มีเครื่องหมายต่อไปนี้ให้เป็นเลขฐานสอง 24 บิตและฐานสิบหกจำนวน 6 หลัก และบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
7	00000000000000000000111	7
8	000000000000000000001000	8
15	000000000000000000001111	F
16	0000000000000000000010000	10
255	0000000000000000011111111	FF
256	0000000000000000100000000	100
65535	0000000111111111111111111	FFFF
65536	0000001000000000000000000	10000

2. จงแปลงเลขฐานสิบต่อไปนี้ให้เป็นเลขฐานสองและฐานสิบหกชนิดมีเครื่องหมายแบบ 2-Complement และบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
+1	0000000000000000000000001	1
-1	1111111111111111111111111	FFFFFF
+15	000000000000000000000001111	F
-16	1111111111111111111110000	FFFF0
+255	0000000000000000011111111	FF
-256	1111111111111111100000000	FFFF00
+65535	0000000111111111111111111	FFFF
-65536	1111111100000000000000000	FF0000

3. จงบวกเลข 2-Complement ต่อไปนี้ แล้วบันทึกผลลัพธ์เป็นฐานสอง ฐานสิบ ข้อผิดพลาดที่แจ้งเตือนและอธิบายเหตุผลว่าทำไมจึงไม่ตรงกัน

- $10000000000000000000000000000000 + 00000000000000000000000000000001$ 
  - ผลลัพธ์ฐานสอง = 100000000000000000000000000000001
  - ผลลัพธ์ฐานสิบหก = 80000001
  - ผลลัพธ์ฐานสิบ = 2147483649
  - ข้อผิดพลาดที่แจ้งเตือน..... Overflow - not enough binary digits to display the number 00000000000000000000000000000001
  - เหตุผล..... เลขเยอะเกินจนไม่สามารถแสดงผลได้
- $10000000000000000000000000000000 + 10000000000000000000000000000000$ 
  - ผลลัพธ์ฐานสอง = 100000000000000000000000000000000
  - ผลลัพธ์ฐานสิบหก = 100000000

- [illegible]

## A.2 การแปลงและคณิตศาสตร์สำหรับมาตรฐาน IEEE754

การทดลองเพื่อให้เข้าใจการแปลงเลขจำนวนจริงฐานสิบให้เป็นเลขฐานสองตามรูปแบบและฝึกการคำนวณโดยใช้คณิตศาสตร์มาตรฐาน IEEE754 Single Precision มีความสอดคล้องกับเนื้อหาในหัวข้อที่ 2.6

### A.2.1 การทดลองสำหรับ Single-Precision

การทดลองนี้จะเน้นที่การแปลงเลขจำนวนจริงให้เป็นเลขฐานสองทศนิยมชนิดลอยตัว สอดคล้องกับเนื้อหาในหัวข้อที่ 2.6 ในรูปแบบ Single Precision โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านถนัด กรอกหรือคลิกที่ชื่อลิงค์ต่อไปนี้

[http://www.binaryconvert.com/convert\\_float.html](http://www.binaryconvert.com/convert_float.html)

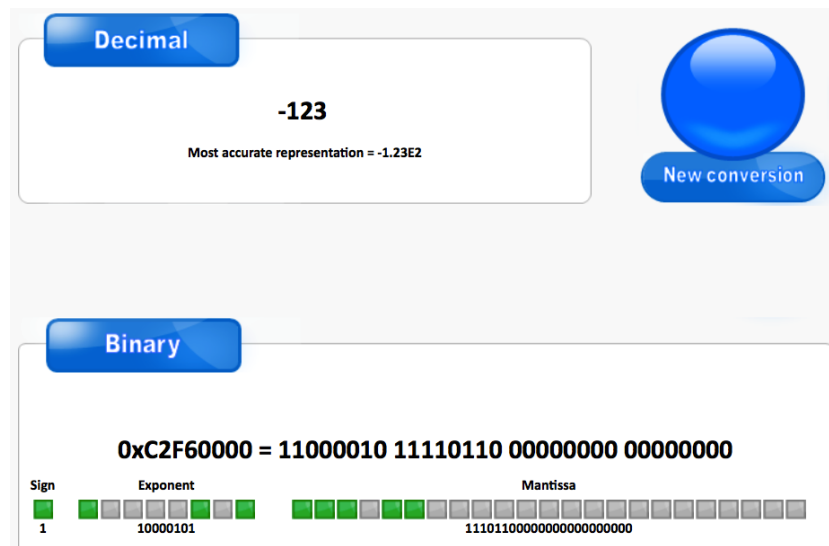
เมื่อเว็บเพจปรากฏขึ้น ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้

1. กรอกเลข 123 ลงในกล่องข้อความ แล้วกดปุ่ม Convert to binary ได้รูปที่ A.6

รูปที่ A.6: ผลลัพธ์จากการแปลงเลข 123.0 ให้เป็นเลขฐานสองชนิด Single Precision

การเรียงตัวของผลลัพธ์เลขฐานสิบหกทางซ้ายมือมาจากเลขฐานสองทางขวามือ ซึ่งเกิดจากบิตข้อมูลทั้งหมด 32 บิตตามรูปแบบของมาตรฐาน IEEE754 ชนิด Single Precision โปรดสังเกต กล่องสี่เหลี่ยมสีเขียวตรงกับบิตที่เป็น '1' กล่องสีเทาตรงกับบิตที่เป็น '0' 0x หมายถึง เลขฐานสิบหก

2. กรอกเลข -123.0 ลงในกล่องข้อความ แล้วกดปุ่ม Convert to binary ได้รูปที่ A.7



รูปที่ A.7: ผลลัพธ์จากการแปลงเลข -123.0 ให้เป็นเลขฐานสองตามมาตรฐาน IEEE754 ชนิด Single Precision

โปรดสังเกตตำแหน่งของกล่องสี่เหลี่ยมหรือสี่เหลี่ยมที่ตรงกับบิต Sign Exponent และ Mantissa ดังนั้น เราจะเห็นได้ว่าเฉพาะ Sign ที่มีการเปลี่ยนแปลง

3. คลิกบนลิงค์นี้ เพื่อทดลองบวกและคูณเลขในรูปแบบ Single Precision ด้วยลิงค์ต่อไปนี้ <http://weitz.de/ieee/> เลื่อนหน้าเว็บลงไปที่ด้านล่างสุด เพื่อค้นหาแถบเมนูตามรูปที่ A.8

binary16    binary32    binary64    binary128

รูปที่ A.8: เมื่อด้านล่างสุดของหน้าเว็บ เพื่อเลือกเลขฐานสองชนิด Single Precision (Binary32) และ Double Precision (Binary64)

4. เลื่อนหน้าเว็บกลับไปด้านบนสุดเพื่อกรอกเลข -123.0 ลงในกล่องข้อความซ้ายบน และ กรอกเลข 123.0 ลงในกล่องข้อความถัดลงมา แล้วกดปุ่ม + แล้วจะได้ผลลัพธ์ดังรูปต่อไปนี้



**IEEE 754 Calculator**  
(See info at bottom of page.)

	Sign	Significand	Exponent
-123.0	1 -	1 . 111011000000000000000000 <b>1.921875</b> 0xC2F60000 0b11000010111101100000000000000000	10000101 <b>+6</b>
123.0	0 +	1 . 111011000000000000000000 <b>1.921875</b> 0x42F60000 0b01000010111101100000000000000000	10000101 <b>+6</b>
<div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <span>+</span> <span>-</span> <span>×</span> <span>/</span> </div>			
0.0	0 +	0 . 000000000000000000000000 <b>0.0</b> 0x00000000 0b00000000000000000000000000000000	00000000 <b>+0</b>

รูปที่ A.9: ผลลัพธ์จากการบวกเลข -123.0+123.0 ให้เป็นเลขฐานสองชนิด Single Precision

จะสังเกตเห็นว่า ผลลัพธ์ที่ได้เรียกว่า True Zero ตามตารางที่ 2.12

5. กดปุ่ม x (คูณ) แล้วจะได้ผลลัพธ์ของ -123×123 ดังรูปต่อไปนี้

**IEEE 754 Calculator**  
(See info at bottom of page.)

	Sign	Significand	Exponent
-123.0	1 -	1 . 111011000000000000000000 <b>1.921875</b> 0xC2F60000 0b11000010111101100000000000000000	10000101 <b>+6</b>
123.0	0 +	1 . 111011000000000000000000 <b>1.921875</b> 0x42F60000 0b01000010111101100000000000000000	10000101 <b>+6</b>
<div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <span>+</span> <span>-</span> <span>×</span> <span>/</span> </div>			
-15129.0	1 -	1 . 110110001100100000000000 <b>1.8468018</b> 0xC66C6400 0b11000110011011000110010000000000	10001100 <b>+13</b>

รูปที่ A.10: ผลลัพธ์จากการคูณเลข -123.0 x 123.0 ให้เป็นเลขฐานสองชนิด Single Precision

จงใช้เว็บเพจลิงค์ต่อไปนี้ในการตอบคำถาม

[illegible]

โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.6 และตรวจคำตอบตามวิธีการทดลองที่ได้ทำไป และกรอกผลลัพธ์ลงบนเส้นประที่จัดไว้ให้เท่านั้น ยกตัวอย่างเช่น

- จงสร้างเลข  $-0.0_{10}$  โดยการทดเลือกบิตที่เปลี่ยนในส่วน Sign เท่านั้น

เลขฐานสอง = 1 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 <sub>2</sub>

ฐานสิบหก = 8 0 0 0 0 0 0 0 <sub>16</sub>
- จงสร้างเลข  $-1.0_{10}$  โดยการทดเลือกบิตที่เปลี่ยนในส่วน Exponent เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 1011 | 1111 | 1000 | 0000 | 0000 | 0000 | 0000 | 0000 <sub>2</sub>

ฐานสิบหก = b f 8 0 0 0 0 0 <sub>16</sub>
- จงสร้างเลข  $-1.5_{10}$  โดยการทดเลือกบิตที่เปลี่ยนในส่วน Mantissa เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 1011 | 1111 | 1100 | 0000 | 0000 | 0000 | 0000 | 0000 <sub>2</sub>

ฐานสิบหก = b f c 0 0 0 0 0 <sub>16</sub>
- จงสร้างเลข  $1.17549435082 \times 10^{-38}$  ซึ่งเป็นค่านอัมัลไลซ์ที่น้อยที่สุด (Normalized)

เลขฐานสอง = 0000 | 0000 | 1000 | 0000 | 0000 | 0000 | 0000 | 0000 <sub>2</sub>

ฐานสิบหก = 0 0 8 0 0 0 0 0 <sub>16</sub>

5. จงสร้างเลข  $5.877472 \times 10^{-39}$  ซึ่งอยู่ในรูป ดีนอร์มัลไลซ์ (Denormalized) เพราะมีค่าน้อยเกินไป

เลขฐานสอง = 0000 | 0000 | 0100 | 0000 | 0000 | 0000 | 0000 | 0000 <sub>2</sub>

ฐานสิบหก = 0 0 4 0 0 0 0 0 <sub>16</sub>
6. จงสร้างเลข  $3.40282346639 \times 10^{38}$  ซึ่งเป็นค่าอนอร์มัลไลซ์ที่มากที่สุด (Normalized)

เลขฐานสอง = 0111 | 1111 | 0111 | 1111 | 1111 | 1111 | 1111 | 1111 <sub>2</sub>

ฐานสิบหก = 7 f 7 f f f f f <sub>16</sub>
7. จงสร้างเลข  $\infty$  ซึ่งเป็นค่าอนันต์ (Infinity) ตามตารางที่ 2.12

เลขฐานสอง = 0111 | 1111 | 1000 | 0000 | 0000 | 0000 | 0000 | 0000 <sub>2</sub>

ฐานสิบหก = 7 f 8 0 0 0 0 0 <sub>16</sub>
8. จงสร้างเลข NaN (Not a Number) ตามตารางที่ 2.12

เลขฐานสอง = 0111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 <sub>2</sub>

ฐานสิบหก = 7 f f f f f f f <sub>16</sub>
9. จงแปลงเลขฐานสองขนาด 32 บิตที่ได้จากกิจกรรมก่อนหน้านี้ ให้เป็น เลขจำนวนเต็ม โดยใช้ลิงค์ต่อไปนี้ [http://www.binaryconvert.com/convert\\_signed\\_int.html](http://www.binaryconvert.com/convert_signed_int.html) เมื่อคัดลอกและวางเลขครบแล้ว ให้กดปุ่ม Convert to decimal

## A.3 รหัสของข้อมูลตัวอักษร

### A.3.1 การทดลอง

การทดลองในหัวข้อนี้จะเป็นการแปลงรหัสตัวอักษรภาษาอังกฤษและไทย เป็นรหัส ASCII และ Unicode ชนิด UCS-2 ตามเนื้อหาในหัวข้อ 2.7 ผ่านทางเว็บไซต์ <https://www.branah.com/ascii-converter> ที่มีการพัฒนาเพื่อเผยแพร่ความรู้เป็นวิทยาทานเช่นเดียวกับเว็บที่ได้ทดลองมา

1. เปิดเว็บตามลิงค์ต่อไปนี้ หรือ กดปุ่มซ้ายบนชื่อลิงค์  
<https://www.branah.com/ascii-converter>
2. กรอกข้อความต่อไปนี้ ลงไปในกล่องข้อความ ASCII  
ไ ท ย ก ข ค a b c  
โปรดสังเกต ระหว่างตัวอักษรมี ช่องว่าง 1 ตัวอักษรเสมอ
3. กดปุ่ม Convert ซ้ายบนสุด จะได้ผลลัพธ์ดังรูปต่อไปนี้

#### ASCII Converter - Hex, decimal, binary, base64, and ASCII converter

Convert ASCII (Example: a b c)

ไ ท ย ก ข ค a b c

Add spaces Remove spaces ☐ Convert white space characters

Convert Hex (Example: 0x61 0x62 0x63) ☒ Remove 0x

e44 e17 e22 e01 e02 e04 61 62 63

Convert Decimal (Example: 97 98 99)

3652 3607 3618 3585 3586 3588 097 098 099

Convert Binary (Example: 01100001 01100010 01100011)

111001000100 111000010111 111000100010 111000000001 111000000010 111000000100 01100001 01100010 01100011

Convert Base64 (Example: YSBiIGM=)

RCAXIClgASACIAQgYSBiIGM=

รูปที่ A.12: ผลลัพธ์จากการกรอกและแปลงตัวอักษร ไ ท ย ก ข ค a b c เป็นรหัสต่างๆ

4. กล่องข้อความ Hex จะแสดงค่า Unicode สำหรับภาษาไทย และ ASCII สำหรับภาษาอังกฤษ ในรูปผู้เขียนได้กดเลือก Remove 0x เพื่อความสะดวกในการอ่านค่า

### A.3.2 กิจกรรมท้ายการทดลอง

1. จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากรหัส ASCII ของตัวอักษร 0 - 9
2. จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากรหัส Unicode ของตัวอักษร ๐ - ๙
3. จงเปิดเว็บที่มีข้อความภาษาไทย เช่น เว็บข่าว แล้วทดลองเปลี่ยนการนำเสนอบนจอเพื่อ View source เช่น Google Chrome ใช้เมนู Tool-> View Source แล้ว Find หรือกดปุ่ม CTRL-F คำว่า charset ว่ามีค่าเท่ากับ utf-8 หรือไม่ เพราะเหตุใด

1.ASCII เป็นเลขฐาน 2 7 bits หาค่าฐาน 10 โดยการแปลงจากฐาน 2 เป็น ฐาน 10 ดังนี้

ตัวอักษร 0 เลขฐาน 2 = 0110000 แปลงเป็น ฐาน 10 =  $2^5 + 2^4 = 48$   
 โดยการนำค่า 0,1 ในแต่ละบิตคูณด้วย  $2^{\text{ตำแหน่งบิต}}$  โดยบิตขวาสุดเป็น 0 บิตถัดมาทางซ้ายเพิ่มไป 1

ตัวอักษร 0 -> ฐาน 10 = 48  
 ตัวอักษร 1 -> ฐาน 10 = 49  
 ตัวอักษร 2 -> ฐาน 10 = 50  
 ตัวอักษร 3 -> ฐาน 10 = 51  
 ตัวอักษร 4 -> ฐาน 10 = 52  
 ตัวอักษร 5 -> ฐาน 10 = 53  
 ตัวอักษร 6 -> ฐาน 10 = 54  
 ตัวอักษร 7 -> ฐาน 10 = 55  
 ตัวอักษร 8 -> ฐาน 10 = 56  
 ตัวอักษร 9 -> ฐาน 10 = 57

2. แปลงฐาน 2 ไปเป็น ฐาน 10 เหมือนข้อที่ 1

ตัวอักษร 0 -> ฐาน 10 = 3663  
 ตัวอักษร 1 -> ฐาน 10 = 3664  
 ตัวอักษร 2 -> ฐาน 10 = 3665  
 ตัวอักษร 3 -> ฐาน 10 = 3666  
 ตัวอักษร 4 -> ฐาน 10 = 3667  
 ตัวอักษร 5 -> ฐาน 10 = 3668  
 ตัวอักษร 6 -> ฐาน 10 = 3669  
 ตัวอักษร 7 -> ฐาน 10 = 3670  
 ตัวอักษร 8 -> ฐาน 10 = 3671  
 ตัวอักษร 9 -> ฐาน 10 = 3672

3.มีค่าเท่ากับ utf-8 เพราะ utf-8 มี Unicode ภาษาไทยอยู่ แต่ ASCII ไม่มีภาษาไทย  
 ดังนั้นจึงจำเป็นต้องใช้ utf-8