

## ภาคผนวก A

# การทดลองที่ 1 ข้อมูลและคณิตศาสตร์ในคอมพิวเตอร์

การทดลองนี้เป็นการทบทวนความเข้าใจและแบบฝึกหัดเสริมของเนื้อหาในบทที่ 2 เนื่องจากจำนวนบิตข้อมูลที่ที่ยาวขึ้นจำเป็นต้องใช้โปรแกรมคอมพิวเตอร์ช่วยคำนวณแทน โดยมีวัตถุประสงค์ ดังต่อไปนี้

- เพื่อให้ เข้าใจ การ แปลง และ คณิตศาสตร์ สำหรับ เลขจำนวน เต็ม ฐาน สอง ชนิด ไม่มี เครื่องหมาย และ มี เครื่องหมายแบบ 2's Complement
- เพื่อให้เข้าใจการแปลงและคณิตศาสตร์สำหรับเลขทศนิยมฐานสองมาตรฐาน IEEE754 ชนิด Single Precision
- เพื่อให้เข้าใจรหัส ASCII และ Unicode สำหรับข้อมูลตัวอักษร

นอกจากเนื้อหาในบทที่ 2 แล้ว ผู้อ่านสามารถศึกษาเว็บเพจเพิ่มเติม เพื่อทำความเข้าใจอย่างลึกซึ้ง ได้แก่

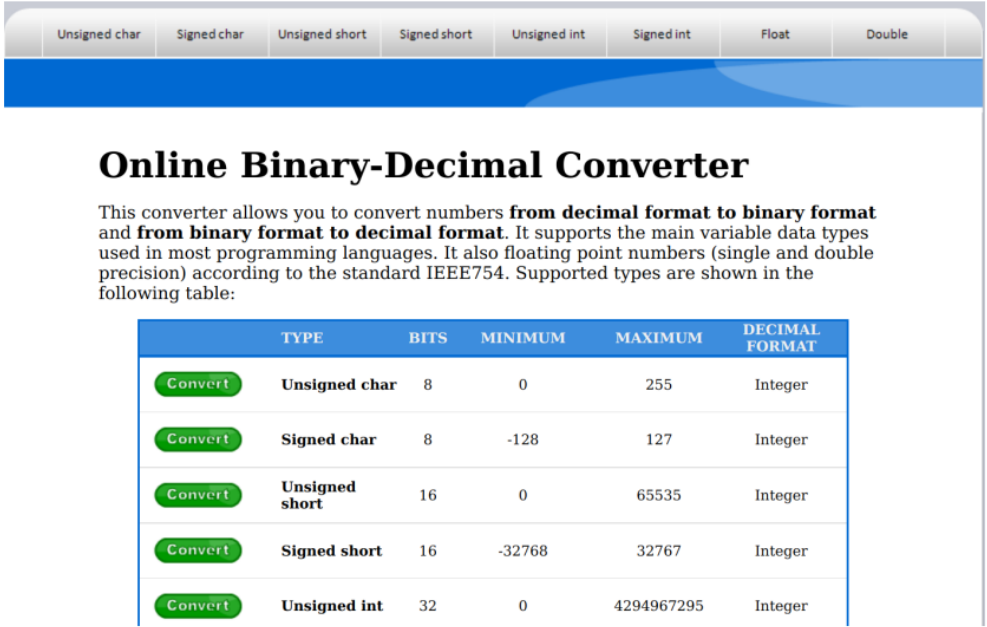
- [https://www.tutorialspoint.com/cprogramming/c\\_data\\_types.htm](https://www.tutorialspoint.com/cprogramming/c_data_types.htm)
- <https://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html>

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

## A.1 การแปลงและคณิตศาสตร์สำหรับเลขจำนวนเต็มฐานสอง

### A.1.1 การทดลองแปลงเลขฐาน

เนื่องจากการแปลงเลขฐานสิบเป็นฐานสองชนิดไม่มีเครื่องหมาย (unsigned) ผู้อ่านสามารถใช้เครื่องคิดเลขทางวิทยาศาสตร์ทั่วไป ดังนั้น การทดลองนี้จะเน้นที่การแปลงเป็นเลขจำนวนเต็มฐานสองชนิดมีเครื่องหมายแบบ 2's Complement สอดคล้องกับเนื้อหาในหัวข้อที่ 2.2 โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านถนัด คลิกที่ชื่อบันทึกต่อไปนี้ <https://www.binaryconvert.com/> ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้



รูปที่ A.1: หน้าเว็บสำหรับแปลงเลขจำนวนเต็มฐานสองเป็นฐานสิบหรือฐานสิบเป็นฐานสองหลายชนิด

1. คลิกที่หัวข้อ Signed Char เพื่อทดลองการแปลงเลขจำนวนเต็มมีเครื่องหมายชนิด 2's Complement ขนาด 8 บิต
2. กรอกเลข -123 ลงในกล่องข้อความ Decimal เพื่อให้โปรแกรมแปลงเลขจำนวนเต็ม -123 เป็นเลขฐานสองมีเครื่องหมายชนิด 2's Complement ดังรูปที่ A.2

รูปที่ A.2: กรอกเลข -123 ลงในกล่องข้อความ Decimal เพื่อให้โปรแกรมแปลงเลขจำนวนเต็ม -123 เป็นเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 2's Complement

กดปุ่ม Convert to binary เพื่อดำเนินการ บันทึกผลลัพธ์ที่ได้จากการแปลงดังต่อไปนี้

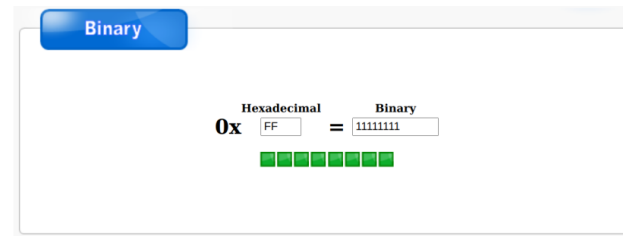
รูปที่ A.3: ผลลัพธ์การแปลงเลข -123 เป็นเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 2's Complement

- Binary (2's Complement) 1 0 0 0 0 1 0 1
- Hexadecimal (0x) 8 5

- แสดงวิธีทำตามสมการที่ (2.16) ที่  $n=8$  บิตเพื่อแปลงเลขให้ตรงตามรูป

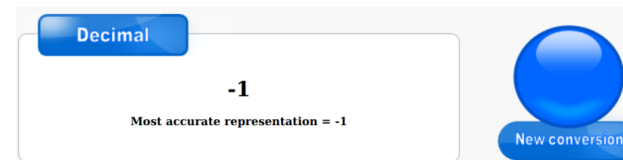
$$\begin{aligned}
 X_{10} &= (-1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\
 &= -128 + 4 + 1 = -123_{10}
 \end{aligned}$$

3. กรอกเลขฐานสองมีเครื่องหมายชนิด 2's Complement 11111111 ขนาด 8 บิตลงในกล่องข้อความ Binary เพื่อให้โปรแกรมแปลงเลขจำนวนเต็มฐานสิบ ดังรูปที่



รูปที่ A.4: การแปลงเลขฐานสองมีเครื่องหมายชนิด 2's Complement 11111111 หรือเท่ากับฐานสิบหก 0xFF

กดปุ่ม Convert to decimal ทางด้านขวาเพื่อดำเนินการ อ่านค่าผลลัพธ์ที่ได้จากการแปลงดังต่อไปนี้



รูปที่ A.5: ผลลัพธ์การแปลงเลขฐานสองมีเครื่องหมายชนิด 2's Complement 11111111 หรือเท่ากับฐานสิบหก 0xFF

4. กดปุ่ม Signed short บนเมนูด้านบนสุด เพื่อเปลี่ยนความยาวเป็น 16 บิต กรอกเลข -123 ลงในกล่องข้อความ Decimal กดปุ่ม Convert to binary เพื่อดำเนินการ บันทึกผลลัพธ์ที่ได้จากการแปลงดังต่อไปนี้

• Binary (2's Complement) 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1

• Hexadecimal (0x) F F 8 5

• แสดงวิธีทำตามสมการที่ (2.16) ที่  $n=16$  บิตเพื่อแปลงเลขให้ตรงตามรูป

$$\begin{aligned}
 x_{10} &= (-1 \times 2^{15}) + (1 \times 2^{14}) + (1 \times 2^{13}) + (1 \times 2^{12}) + (1 \times 2^{11}) + \dots + (1 \times 2^8) \\
 &\quad + (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\
 &= -32,768 + 16,384 + \dots + 128 + 0 + 0 + 0 + 0 + 4 + 0 + 1 \\
 &= -123_{10}
 \end{aligned}$$



A.1.2 คณิตศาสตร์เลขจำนวนเต็มฐานสอง

1. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมาย  
ชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์  $V$

		$c_8$	$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$	$V=c_8 \oplus c_7$
		<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	$V=\underline{1} \oplus \underline{1}$
$X$	-123		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	
$+Y$	+ -1	+	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	
$Z$	<u>-124</u>		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	

ซอฟต์แวร์สามารถนำผลลัพธ์  $Z$  ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด **9 ใช้งานได้เนื่องจาก  $V=0$  แสดงว่า ไม่เกิด Overflow**

2. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมาย  
ชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์  $V$

		$c_8$	$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$	$V=c_8 \oplus c_7$
		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	0	$V=\underline{1} \oplus \underline{0}$
$X$	-123		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	
$+Y$	+ -123	+	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<b>overflow</b>
$Z$	<u>10</u>		<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	

ซอฟต์แวร์สามารถนำผลลัพธ์  $Z$  ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด **9 ใช้งานไม่ได้เนื่องจาก  $V=1$  แสดงว่า เกิด Overflow & สังเกตได้จาก 3๓๓๓: 1๑๑๑ bit ที่สามารถเก็บได้**

3. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมาย  
ชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์  $V$

		$c_8$	$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$	$V=c_8 \oplus c_7$
		<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	0	$V=\underline{0} \oplus \underline{0}$
$X$	-123		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	
$+Y$	+ 1	+	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	
$Z$	<u>-122</u>		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	

ซอฟต์แวร์สามารถนำผลลัพธ์  $Z$  ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด **9 ใช้งานได้เนื่องจาก  $V \neq 0$  แสดงว่า ไม่เกิด Overflow**

4. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมาย  
ชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์  $V$

		$c_8$	$c_7$	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$	$V=c_8 \oplus c_7$
		<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	0	$V=\underline{1} \oplus \underline{1}$
$X$	-123		<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	
$+Y$	+ 123		<u>0</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	
$Z$	<u>+0</u>		<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	

ซอฟต์แวร์สามารถนำผลลัพธ์  $Z$  ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด **9 ใช้งานได้เนื่องจาก  $V \neq 0$  แสดงว่า ไม่เกิด Overflow**

## A.1.3 กิจกรรมท้ายการทดลอง

จงทำการทดลองและตอบคำถามต่อไปนี้ โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.2.2 และตรวจคำตอบตามวิธีการทดลองที่ได้ทำไป

1. จงแปลงเลขจำนวนเต็มฐานสิบชนิดไม่มีเครื่องหมายต่อไปนี้ให้เป็นเลขจำนวนเต็มฐานสอง 16 บิตและฐานสิบหกจำนวน 4 หลัก และบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
7	0000 0000 0000 0001 11 <sub>2</sub>	__ 0007 <sub>16</sub>
8	0000 0000 0000 0010 00 <sub>2</sub>	__ 0008 <sub>16</sub>
15	0000 0000 0000 0011 11 <sub>2</sub>	__ 000F <sub>16</sub>
16	0000 0000 0000 0100 00 <sub>2</sub>	__ 0010 <sub>16</sub>
255	0000 0000 0011 1111 11 <sub>2</sub>	__ 00FF <sub>16</sub>
256	0000 0000 0100 0000 00 <sub>2</sub>	__ 0100 <sub>16</sub>
65535	0111 1111 1111 1111 11 <sub>2</sub>	__ FFFF <sub>16</sub>
65536	1000 0000 0000 0000 00 <sub>2</sub>	__ 0000 <sub>16</sub>

} ไม่สามารถแปลงได้  
ขงทดเกิน 16 bits

2. จงแปลงเลขจำนวนเต็มฐานสิบต่อไปนี้ให้เป็นเลขจำนวนเต็มฐานสองและฐานสิบหกชนิดมีเครื่องหมายแบบ 2's Complement ความยาว 16 บิตแล้วบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
+1	0000 0000 0000 0001 <sub>2</sub>	__ 0001 <sub>16</sub>
-1	1111 1111 1111 1111 <sub>2</sub>	__ FFFF <sub>16</sub>
+15	0000 0000 0000 0111 <sub>2</sub>	__ 000F <sub>16</sub>
-16	1111 1111 1111 0000 <sub>2</sub>	__ FFFF <sub>16</sub>
+255	0000 0000 1111 1111 <sub>2</sub>	__ 00FF <sub>16</sub>
-256	1111 1111 0000 0000 <sub>2</sub>	__ FFFF <sub>16</sub>
+65535	1111 1111 1111 1111 <sub>2</sub>	__ FFFF <sub>16</sub>
-65536	0000 0000 0000 0000 <sub>2</sub>	__ 0000 <sub>16</sub>

3. จงบวกเลข 2's Complement ต่อไปนี้ แล้วบันทึกผลลัพธ์เป็นฐานสองความยาว 16 บิต ฐานสิบหก ฐานสิบ โอเวอร์โฟลว์หรือไม่ และอธิบายเหตุผลว่าทำไมจึงไม่ตรงกัน

- 1000000000000000 + 0000000000000001

– ผลลัพธ์ = 1000 0000 0000 0001<sub>2</sub>

– ผลลัพธ์ = 8001<sub>16</sub>

– ผลลัพธ์ = 32 769<sub>10</sub>

– โอเวอร์โฟลว์หรือไม่... ไม่ Overflow

– เหตุผล... V = 0, C<sub>16</sub> = 0 และ ได้ 0 สามารถเก็บได้พอดี 16 bits

- 1000000000000000 + 1000000000000000

– ผลลัพธ์ = 1000 0000 0000 0000<sub>2</sub>

– ผลลัพธ์ = 10000<sub>16</sub>

} เก็บ 16 bits

- ผลลัพธ์ =  $\underline{65536}_{10}$
  - โอเวอร์โฟลว์หรือไม่... Overflow
  - เหตุผล  $V = C_{16} \oplus C_{15}$  แล้วได้ 1 จะเกิน 16 bits
- $10000000000000000 - 00000000000000001 \rightarrow 1000\ 0000\ 00000000_2 + 1111\ 1111\ 1111\ 1111$   
 $= 1\ 0111\ 1111\ 1111\ 1111_2$  (เอาตัวลบมา 2' แล้ว + 1)
- ผลลัพธ์ =  $\underline{0111111111111111}_2$
  - ผลลัพธ์ =  $\underline{7FFF}_{16}$
  - ผลลัพธ์ =  $\underline{-32768}_{10}$
  - โอเวอร์โฟลว์หรือไม่... เกิด Overflow
  - เหตุผล  $V = C_{16} \oplus C_{15}$  แล้วได้ 1 จะเกิน 16 bits
- $10000000000000000 - 10000000000000000$
- ผลลัพธ์ =  $\underline{0000000000000000}_2$
  - ผลลัพธ์ =  $\underline{0000}_{16}$
  - ผลลัพธ์ =  $\underline{0}_{10}$
  - โอเวอร์โฟลว์หรือไม่... ไม่เกิด Overflow
  - เหตุผล  $V = C_{16} \oplus C_{15}$  แล้วได้ 0 จะไม่เกิน 16 bits



## A.2 การแปลงและคณิตศาสตร์เลขทศนิยมฐานสองมาตรฐาน IEEE754

การทดลองเพื่อให้เข้าใจการแปลงเลขทศนิยมฐานสิบให้เป็นเลขฐานสองตามรูปแบบและฝึกการคำนวณโดยใช้คณิตศาสตร์มาตรฐาน IEEE754 Single Precision มีความสอดคล้องกับเนื้อหาในหัวข้อที่ 2.6

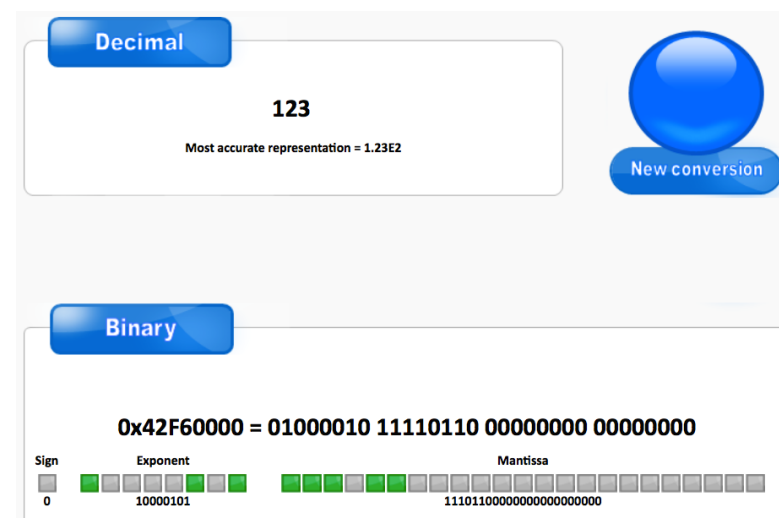
### A.2.1 เลขทศนิยมชนิดจุดลอยตัวมาตรฐาน IEEE754 Single-Precision

การทดลองนี้จะเน้นที่การแปลงเลขทศนิยมฐานสิบให้เป็นเลขทศนิยมฐานสองชนิดจุดลอยตัว สอดคล้องกับเนื้อหาในหัวข้อที่ 2.6 ในรูปแบบ Single Precision โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านถนัด คลิกที่ชื่อลิงก์ต่อไปนี้

[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 ให้เป็นเลขทศนิยมฐานสองชนิด Single Precision

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

แสดงวิธีทำตามสมการที่ (2.67) เพื่อแปลงเลขให้ตรงตามรูป

[0][10000101][111011000000000000000000]

↓                      ↓                      ↓  
S                      Exponent                      Floating

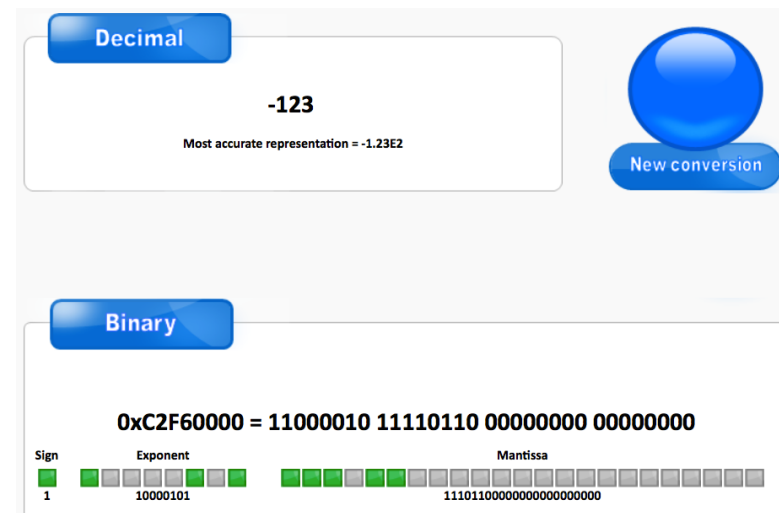
Sign bit = 0

Exponent =  $1000\ 0101_2 = 133$

$E_{IEEE} = 133$ ,  $E_{float} = 127$

$$\begin{aligned}
 F_{10, IEEE} &= (-1)^S \times \left[ 1 + \frac{Y-1}{2} + \frac{Y-2}{4} + \frac{Y-3}{8} + \frac{Y-4}{16} + \frac{Y-5}{32} + \frac{Y-6}{64} \right] \times 2^{E_{IEEE} - E_{float}} \\
 &= (-1)^0 \times \left[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + 0 + \frac{1}{32} + \frac{1}{64} \right] \times 2^{133 - 127} \\
 &= \left[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} + \frac{1}{64} \right] \times 64 \\
 &= 64 + 32 + 16 + 8 + 2 + 1 \\
 &= 123
 \end{aligned}$$

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



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

โปรดสังเกตตำแหน่งของกล่องสี่เหลี่ยมหรือสี่เหลี่ยมที่ตรงกับบิต Sign Exponent และ Mantissa ดังนั้น เราจะเห็นได้ว่าเฉพาะ Sign ที่มีการเปลี่ยนแปลง

แสดงวิธีทำตามสมการที่ (2.67) เพื่อแปลงเลขให้ตรงตามรูป

[1] [1000 0101] [1110 1100 0000 0000 0000]  
 ↓ ↓ ↓  
 S Exponent Floating

$$\text{Sign bit} = 1$$

$$\text{Exponent} = 1000\ 0101_2 = 133$$

$$E_2, \text{IEEE} = 133, E_{\text{bias}} = 127$$

$$\begin{aligned} F_{10, \text{IEEE}} &= (-1)^1 \times \left[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + 0 + \frac{1}{32} + \frac{1}{64} \right] \times 2^{133-127} \\ &= (-1) \left[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + 0 + \frac{1}{32} + \frac{1}{64} \right] \times 64 \\ &= (-1) [64 + 32 + 16 + 8 + 2 + 1] \\ &= -123 \end{aligned}$$

3. คลิกบนลิงก์นี้ เพื่อทดลองบวกและคูณเลขในรูปแบบ Single Precision ด้วยลิงก์ต่อไปนี้ <http://weitz.de/ieee/> เลื่อนหน้าเว็บลงไปด้านล่างสุด เพื่อค้นหาแถบเมนูตามรูปที่ A.8 แล้วกดเลือกปุ่ม binary32 เพื่อทดลองการบวกและคูณเลข IEEE754 Single Precision

binary16 binary32 binary64 binary128

รูปที่ A.8: เมนูด้านล่างสุดของหน้าเว็บ เพื่อเลือกเลขทศนิยมฐานสองชนิด IEEE754 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 1.921875 0xC2F60000 0b11000010111101100000000000000000	10000101 +6
123.0	0 +	1.111011000000000000000000 1.921875 0x42F60000 0b01000010111101100000000000000000	10000101 +6
+ - × /			
0.0	0 +	0.000000000000000000000000 0.0 0x00000000 0b00000000000000000000000000000000	00000000 +0

รูปที่ A.9: ผลลัพธ์จากการบวกเลข -123.0+123.0 ให้เป็นเลขทศนิยมฐานสองชนิด IEEE754 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-top: 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 ให้เป็นเลขทศนิยมฐานสองชนิด IEEE754 Single Precision

แสดงวิธีทำตามสมการที่ (2.67) เพื่อแปลงเลขให้ตรงตามผลคูณในรูปที่ A.10

$$\text{ผลคูณ} = -123 \times 123 = -15129$$

$$[1] [1000\ 1100] [1101\ 1000\ 0110\ 0100\ 0000\ 000]$$

↓      ↓      ↓  
S      Exponent      Floating

$$\text{Sign bit} = 1$$

$$\text{Exponent} = 1000\ 1100_2 = 140$$

$$E_{2, \text{IEEE}} = 140, E_{\text{bias}} = 127$$

$$F_{10, \text{IEEE}} = (-1)^5 \times \left[ 1 + \frac{y^{-1}}{2} + \frac{y^{-2}}{4} + \frac{y^{-3}}{8} + \dots + \frac{y^{-8}}{256} + \frac{y^{-9}}{512} + \frac{y^{-10}}{1024} \right] \times 2^{140-127}$$

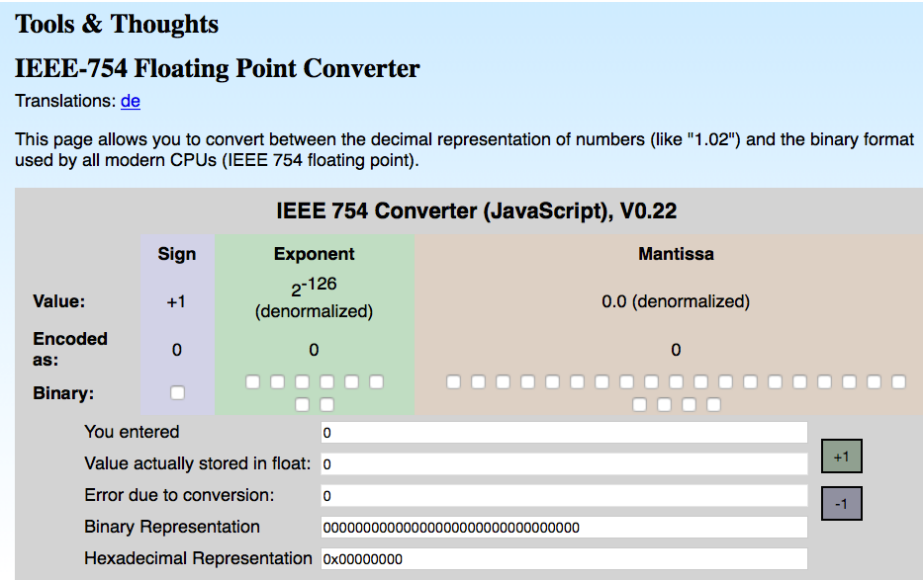
$$= (-1) \left[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{32} + \frac{1}{512} + \frac{1}{1024} \right] \times 2^{19}$$

$$= (-1) [8192 + 4096 + 2048 + 512 + 256 + 16 + 8 + 1]$$

$$= -15129$$

A.2.2 กิจกรรมท้ายการทดลอง

จงใช้ลิงก์ของเว็บเพจต่อไปนี้ในการตอบคำถาม  
<https://www.h-schmidt.net/FloatConverter/IEEE754.html>



รูปที่ A.11: เว็บสำหรับการตอบคำถามเพื่อสร้างเลขหรือแปลงเลขฐานสิบด้วยมาตรฐาน IEEE754 Single Precision การกดเลือกคือทำให้บิตนั้นเท่ากับ '1'

โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.6 และตรวจคำตอบตามวิธีการทดลองที่ได้ทำไป และบันทึกผลลัพธ์ลงบนเส้นประที่จัดไว้ให้เท่านั้น ผู้อ่านสามารถกดเปลี่ยนเครื่องหมายถูก ซึ่งแทนลอจิก 1 หากไม่มีเครื่องหมายถูกแทนลอจิก 0 ยกตัวอย่างเช่น

1. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-0.0_{10}$  โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน 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>  
ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....-0.....  
ความผิดพลาด (Error due to conversion).....\.....
2. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-1.0_{10}$  โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน Exponent เท่านั้น ต่อจากข้อที่แล้ว  
เลขฐานสอง = 1 0 1 1|1 1 1 1|1 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 = B F 8 0 0 0 0 0<sub>16</sub>  
ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....-1.....  
ความผิดพลาด (Error due to conversion).....\.....
3. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-1.55_{10}$  หรือ  $1.55e0$  โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน Mantissa เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 1011|1111|1100|0110|0110|0110|0110|0110 <sub>2</sub>  
 ฐานสิบหก = B F C 6 6 6 6 6 6 <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **1.549999523162841796875**  
 ความผิดพลาด (Error due to conversion): **4.26337152302125E-8**

4. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $1.17549435082 \times 10^{-38}$  หรือ  $1.17549435082\text{e-}38$  ซึ่งเป็นค่านอร์มัลไลซ์ที่น้อยที่สุด (Normalize)

เลขฐานสอง = 0000|0000|1000|0000|0000|0000|0000|0000 <sub>2</sub>  
 ฐานสิบหก = 0 0 8 0 0 0 0 0 0 <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **1.17549435.....**  
 ความผิดพลาด (Error due to conversion): **2.28750796E-50**

5. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $1.17549421069 \times 10^{-38}$  หรือ  $1.17549421069\text{e-}38$  ซึ่งอยู่ในรูป ดินอร์มัลไลซ์ (Denormalize) เพราะมีค่าน้อยกว่าค่านอร์มัลไลซ์ที่ต่ำที่สุด

เลขฐานสอง = 0000|0000|0111|1111|1111|1111|1111|1111 <sub>2</sub>  
 ฐานสิบหก = 0 0 4 f f f f f f <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **1.1754942106924410754670214443442370824287458133185717**  
 ความผิดพลาด (Error due to conversion): **2.441075437021444344237148827E-50**

6. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $1.40129846432 \times 10^{-45}$  หรือ  $1.40129846432\text{e-}45$  ซึ่งอยู่ในรูป ดินอร์มัลไลซ์ (Denormalize) และต่ำที่สุด

เลขฐานสอง = 0000|0000|0000|0000|0000|0000|0000|0001 <sub>2</sub>  
 ฐานสิบหก = 0 0 0 0 0 0 0 1 <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **1.4012984643248170709217295823**  
 ความผิดพลาด (Error due to conversion): **4.8170709237E-57**

7. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $1.0 \times 10^{-46}$  หรือ  $1\text{e-}46$  ซึ่งอยู่ในรูป ดินอร์มัลไลซ์ (Denormalize) และจัดเก็บด้วยค่า 0.0 แทน

เลขฐานสอง = 0000|0000|0000|0000|0000|0000|0000|0000 <sub>2</sub>  
 ฐานสิบหก = 0 0 0 0 0 0 0 0 <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **0.....**  
 ความผิดพลาด (Error due to conversion): **-1E-46.....**

8. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $3.40282346640 \times 10^{38}$  หรือ  $3.40282346640\text{e}38$  ซึ่งเป็นค่านอร์มัลไลซ์ที่มากที่สุด

เลขฐานสอง = 0111|1111|0111|1111|1111|1111|1111|1111 <sub>2</sub>  
 ฐานสิบหก = 7 f 7 f f f f f <sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float): **3.40282346638528859811704183484616425440**

ความผิดพลาด (Error due to conversion)...-14711401882958165154 83074560

9. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $3.5 \times 10^{38}$  หรือ  $3.5e+38$  ซึ่งมากกว่าค่าอนอร์มัลไลซ์ที่มากที่สุด ซึ่งหมายถึงค่าอนันต์ ( $\infty$ : Infinity) ตามตารางที่ 2.12

เลขฐานสอง = 0 1 1 1 | 1 1 1 1 | 1 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 = 7 f 8 0 0 0 0 <sub>16</sub>

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float)...Infinity.....

ความผิดพลาด (Error due to conversion)...Infinity.....

10. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ NaN (Not a Number) ตามตารางที่ 2.12

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

ฐานสิบหก = 7 f f f f f f f <sub>16</sub>

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float)...NaN.....

ความผิดพลาด (Error due to conversion)...-.....

(9 เ็จไม่เซียบ)

เมื่อ expo ต่ำสุด 0 และ bit 16 เป็น 1 หรือมากกว่า 1

## 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> ที่มึนั้กพัฒนาเพื่อเผยแพร่ความรู้เป็นวิทยาทานเช่นเดียวกับเว็บที่ได้ทดลองมา

1. เปิดเว็บตามลิงก์ต่อไปนี้ หรือ กดปุ่มซ้ายบนชื่อลิงก์  
<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)

ไทย 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

- จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากระหัส ASCII ของตัวอักษร 0 - 9
- จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากระหัส Unicode ของตัวอักษร ๐ - ๙
- 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

Char set ที่อยู่ใน View Source มีค่าเท่ากับ UTF-8

โดยปกติตัวอักษร ASCII ในตาราง ๐-๒๕๕ จะใช้ ๑ byte ก็เพียงพอต่อการนำไปใช้งาน แต่ในภาษาอื่นด้วย ก็มีความจำเป็นที่จะต้องใช้ byte มากกว่า ๑ byte ซึ่งรวมไปถึงภาษาไทยด้วย ในภาษาไทยนั้นตัวอักษร + สระ + วรรณยุกต์ มีจำนวนเกินกว่า ๑ byte จะสามารถระบุได้ถึงตัวอักษรตามยาวเพิ่ม

ASCII ของ 0-9 ได้จาก 48 ถึง 57 โดยวิธีการหาฐาน 10 จาก ASCII ที่ 0 ตรงๆ ASCII number ที่อยู่ระหว่าง 48-57 ลง ๐๐๐ ลง 48 เช่น ASCII number 50 ลง ๐๐๐ ลง 48 จะได้ ๑ ก็คือ เลข ๑ ในฐาน 10

Unicode number 0-๙ คือ 3๖๖4 - ๓๖71 วิธีการหาฐาน 10 คือตรงๆคือ Unicode number ลง ๐๐๐ ลง 3๖๖4 เช่น ๓ (เลข ๓) มี Unicode คือ 3๖๖6 ลง ๐๐๐ ลง 3๖๖4 จะได้ ๒ หรือ ๒ ในฐาน 10