

## ภาคผนวก A

# การทดลองที่ 1 ข้อมูลและคณิตศาสตร์ในคอมพิวเตอร์

การทดลองนี้เป็นการทบทวนความเข้าใจและแบบฝึกหัดเสริมของเนื้อหาในบทที่ 2 เนื่องจากจำนวนบิตข้อมูลที่ยาวขึ้นจำเป็นต้องใช้โปรแกรมคอมพิวเตอร์ช่วยคำนวณแทน โดยมีวัตถุประสงค์ ดังต่อไปนี้

- เพื่อให้เข้าใจ 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/daterepresentation.html>

ผู้อ่านจะพบว่าเนื้อหาในเว็บของมหาวิทยาลัยนันทยา 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 คลิกที่ชื่อลิงก์ต่อไปนี้ <https://www.binaryconvert.com/> ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้

TYPE	BITS	MINIMUM	MAXIMUM	DECIMAL FORMAT
<a href="#">Convert</a> Unsigned char	8	0	255	Integer
<a href="#">Convert</a> Signed char	8	-128	127	Integer
<a href="#">Convert</a> Unsigned short	16	0	65535	Integer
<a href="#">Convert</a> Signed short	16	-32768	32767	Integer
<a href="#">Convert</a> Unsigned int	32	0	4294967295	Integer

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

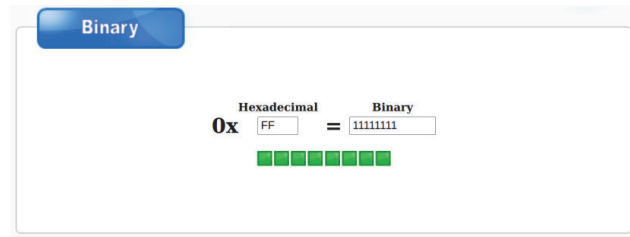
รูปที่ A.3: ผลลัพธ์การแปลงเลข -123 เป็นเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 2's Complement

- Binary (2's Complement) 10000101
- Hexadecimal (0x) 85
- แสดงวิธีทำตามสมการที่ (2.16) ที่  $n=8$  บิตเพื่อแปลงเลขให้ตรงตามรูป

$$x_{2,8} = 1000\ 0101_2 = 85_{16}$$

$$\begin{aligned} x_{10,8} &= -1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= -128 + 4 + 1 \\ &= -123_{10} \end{aligned}$$

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) 111111110000101
- Hexadecimal (0x) FF85
- แสดงวิธีทำตามสมการที่ (2.16) ที่  $n=16$  บิตเพื่อแปลงเลขให้ตรงตามรูป

$$\begin{aligned}
 x_{10,8} &= -1 \times 2^{15} + 1 \times 2^{14} + \dots + 1 \times 2^7 + 0 \times 2^6 + \dots + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= -32,768 + \dots + 128 + 4 + 1 \\
 &= -123_{10}
 \end{aligned}$$

5. กดปุ่ม Signed int บนเมนูด้านบนสุด เพื่อเปลี่ยนความยาวเป็น 32 บิต กรอกเลข -123 ลงในกล่องข้อความ Decimal กดปุ่ม Convert to binary เพื่อดำเนินการ บันทึกผลลัพธ์ที่ได้จากการแปลงดังต่อไปนี้

- Binary 1111111111111111111111110000101
- Hexadecimal (0x) FFFF85
- แสดงวิธีทำตามสมการที่ (2.16) ที่  $n=32$  บิตเพื่อแปลงเลขให้ตรงตามรูป

$$\begin{aligned} x_{10,5} &= -1 \times 2^{31} + 1 \times 2^{30} + \dots + 1 \times 2^7 + 0 \times 2^6 + \dots + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= -2,147,483,648 + \dots + 128 + 4 + 1 \\ &= -123_{10} \end{aligned}$$

## A.1.2 คณิตศาสตร์เลขจำนวนเต็มฐานสอง

1. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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 ใช้ได้ เพราะ ไม่เกิด Overflow ค่าตัวจำนวน  $1 \oplus 1 = 0$  ซอฟต์แวร์จึงนำไปใช้ได้

2. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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>	
$Z$	<u>-246</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 ใช้ไม่ได้ เพราะ เกิด Overflow ค่าตัวจำนวน  $1 \oplus 0 = 1$  ซอฟต์แวร์จึงไม่สามารถนำไปใช้ได้

3. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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 ใช้ได้ เพราะ ไม่เกิด Overflow ค่าตัวจำนวน  $0 \oplus 0 = 0$  ซอฟต์แวร์จึงนำไปใช้ได้

4. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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 ใช้ได้ เพราะ ไม่เกิด Overflow ค่าตัวจำนวน  $1 \oplus 1 = 0$  ซอฟต์แวร์จึงนำไปใช้ได้

## A.1.3 กิจกรรมท้ายการทดลอง

จงทำการทดลองและตอบคำถามต่อไปนี้ โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.2.2 และตรวจคำตอบตามวิธีการทดลองที่ได้ทำไป

1. จงแปลงเลขจำนวนเต็มฐานสิบชนิดไม่มีเครื่องหมายต่อไปนี้ให้เป็นเลขจำนวนเต็มฐานสอง 16 บิตและฐานสิบหกจำนวน 4 หลัก และบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
7	0000000000000111 <sub>2</sub>	__ 0007 <sub>16</sub>
8	0000000000001000 <sub>2</sub>	__ 0008 <sub>16</sub>
15	0000000000001111 <sub>2</sub>	__ 000F <sub>16</sub>
16	0000000000010000 <sub>2</sub>	__ 0010 <sub>16</sub>
255	0000000011111111 <sub>2</sub>	__ 00FF <sub>16</sub>
256	0000000100000000 <sub>2</sub>	__ 0100 <sub>16</sub>
65535	1111111111111111 <sub>2</sub>	__ FFFF <sub>16</sub>
65536	1111111111111111 <sub>2</sub>	__ FF00 <sub>16</sub>

→ ค่ามากที่สุดคือ 65535  
( รับ 16 บิต )

2. จงแปลงเลขจำนวนเต็มฐานสิบต่อไปนี้ให้เป็นเลขจำนวนเต็มฐานสองและฐานสิบหกชนิดมีเครื่องหมายแบบ 2's Complement ความยาว 16 บิตแล้วบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
+1	0000000000000001 <sub>2</sub>	__ 0001 <sub>16</sub>
-1	1111111111111111 <sub>2</sub>	__ FFFF <sub>16</sub>
+15	0000000000001111 <sub>2</sub>	__ 000F <sub>16</sub>
-16	1111111111110000 <sub>2</sub>	__ FFFF <sub>16</sub>
+255	0000000011111111 <sub>2</sub>	__ 00FF <sub>16</sub>
-256	1111111100000000 <sub>2</sub>	__ FF00 <sub>16</sub>
+65535	0111111111111111 <sub>2</sub>	__ 7FFF <sub>16</sub>
-65536	1000000000000000 <sub>2</sub>	__ 8000 <sub>16</sub>

→ ค่ามากที่สุดคือ 32767  
→ ค่าน้อยที่สุดคือ -32768

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

- $1000000000000000 + 0000000000000001$

- ผลลัพธ์ =  $1\ 0000000000000001_2$

- ผลลัพธ์ =  $8001_{16}$

- ผลลัพธ์ =  $-32767_{10}$

- โอเวอร์โฟลว์หรือไม่...  $1_2$

- เหตุผล...  $2_{16} \oplus 15_{16} = 0 \oplus 0 = 0$  พอล์แวร์จึงน่าจะได้

- $1000000000000000 + 1000000000000000$

- ผลลัพธ์ =  $0000000000000000_2$

- ผลลัพธ์ =  $0000_{16}$

- ผลลัพธ์ =  $\underline{00000}_{10}$
- โอเวอร์โฟลว์หรือไม่.....เกิด overflow
- เหตุผล..... $\underset{16}{C} \oplus \underset{15}{C} = 1 \oplus 0 = 1$  ซอฟต์แวร์จึงไม่สามารถใช้งานได้
- $1000000000000000 - 0000000000000001$ 
  - ผลลัพธ์ =  $\underline{01111111111111}_{2}$
  - ผลลัพธ์ =  $\underline{FFFF}_{16}$
  - ผลลัพธ์ =  $\underline{32767}_{10}$
  - โอเวอร์โฟลว์หรือไม่.....เกิด overflow
  - เหตุผล..... $\underset{16}{C} \oplus \underset{15}{C} = 1 \oplus 0 = 1$  ซอฟต์แวร์จึงไม่สามารถใช้งานได้
- $1000000000000000 - 1000000000000000$ 
  - ผลลัพธ์ =  $\underline{0000000000000000}_{2}$
  - ผลลัพธ์ =  $\underline{0000}_{16}$
  - ผลลัพธ์ =  $\underline{00000}_{10}$
  - โอเวอร์โฟลว์หรือไม่.....เกิด overflow
  - เหตุผล..... $\underset{16}{C} \oplus \underset{15}{C} = 1 \oplus 0 = 0$  ซอฟต์แวร์จึงไม่สามารถใช้งานได้



## 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 โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านกดคลิกที่ชื่อลิงก์ต่อไปนี้

[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

$$42F60000_{16} = [0100 \ 0010 \ 1111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000]_2$$

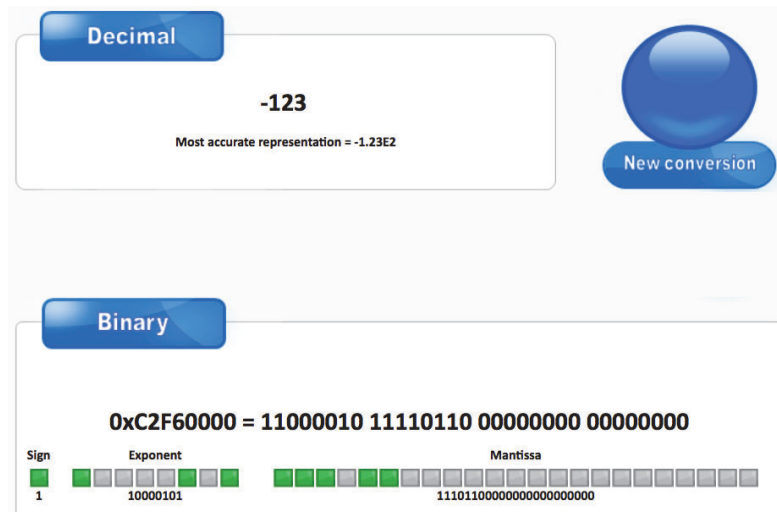
$$s = [0] [E_{2,IEEE} = 100 \ 0010 \ 1] [Y_2 = 111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000]_2$$

$$E_{true} = 1000 \ 0101_2 - E_{bias} = 133 - 127 = 6$$

$$Y_2 = 111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000_2$$

$$\begin{aligned} F_{10,IEEE} &= (-1)^s (1 + .111011_2) \times 2^6 \\ &= 1 \times (1.111011_2) \times 2^6 \\ &= 1111011_2 = 4 + 32 + 16 + 8 + 2 + 1 = 123.0_{10} \end{aligned}$$

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 เราจะเห็นว่าเฉพาะ Sign ที่มีการเปลี่ยนแปลง

แสดงวิธีทำตามสมการที่ (2.67) เพื่อแปลงเลขให้ตรงตามรูป

$$C2F60000_{16} = [1100 \ 0010 \ 1111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000]_2$$

$$S = [1][E_{2,IEEE} = 100 \ 0010 \ 1][Y_2 = 111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000]_2$$

$$E_{true} = 1000 \ 0101_2 - E_{bias} = 133 - 127 = 6$$

$$Y_2 = 111 \ 0110 \ 0000 \ 0000 \ 0000 \ 0000$$

$$\begin{aligned} F_{10,IEEE} &= (-1)^1 \times (1 + .111011_2) \times 2^6 \\ &= (-1) \times (1.111011_2) \times 2^6 \end{aligned}$$

$$= (-1) \times (1111011_2)$$

$$= -123.0_{10}$$

3. คลิกบนลิงก์นี้ เพื่อทดลองบวกและคูณเลขในรูปแบบ Single Precision ด้วยลิงก์ต่อไปนี้ <http://weitz.de/ieee/> เลื่อนหน้าเว็บลงไปที่ด้านล่างสุด 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
<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 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 × (คูณ) แล้วจะได้ผลลัพธ์ของ -123×123 ดังรูปต่อไปนี้

**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
<div> <div>+</div> <div>-</div> <div>x</div> <div>/</div> </div>			
-15129.0	1 -	1 . 110110001100100000000000 1.8468018 0xC66C6400 0b11000110011011000110010000000000	10001100 +13

รูปที่ A.10: ผลลัพธ์จากการคูณเลข -123.0 x 123.0 ให้เป็นเลขทศนิยมฐานสองชนิด IEEE754 Single Precision

แสดงวิธีทำตามสมการที่ (2.67) เพื่อแปลงเลขให้ตรงตามผลคูณในรูปที่ A.10

$$S = 1$$

$$E_{\text{true}} = 10001100 - E_{\text{bias}} = 140 - 127 = 13$$

$$M_2 = 1101100011001$$

$$\begin{aligned}
 F_{10, \text{IEEE}} &= (-1)^1 \times (1 + .1101100011001_2) \times 2^{13} \\
 &= (-1) \times (11101100011001_2) \\
 &= -15129.0_{10}
 \end{aligned}$$

### A.2.2 กิจกรรมท้ายการทดลอง

จงใช้ลิงก์ของเว็บเพจต่อไปนี้ในการตอบคำถาม

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

[illegible]

**รูปที่ A.11:** เว็บบสำหรับการตอบคำถามเพื่อสร้างเลขหรือแปลงเลขฐานสิบด้วยมาตรฐาน IEEE754 Single Precision การกดเลือกคือทำให้บิตนั้นเท่ากับ '1'

โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.6 และตรวจคำตอบตามวิธีทำการทดลองที่ได้ทำไป และบันทึกผลลัพธ์ลงบนเส้นประที่จัดไว้ให้เท่านั้น ผู้อ่านสามารถดัดแปลงเครื่องหมายถูก ซึ่งแทนลอจิก 1 หากไม่มีเครื่องหมายถูกแทนลอจิก 0 ยกตัวอย่างเช่น

1. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-0.0_{10}$  โดยการกดเลือกปุ่มสีเหลี่ยมในส่วน 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).....0

2. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-1.0_{10}$  โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน Expo-  
nent เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 10 11 11 11 1000 0000 0000 0000 0000 0000<sub>2</sub>

ฐานสิบหก = B F 6 0 0 0 0 0<sub>16</sub>

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....-1.....

ความผิดพลาด (Error due to conversion)..... 0,0

3. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ  $-1.55_{10}$  หรือ  $1.55e0$  โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน Mantissa เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 01111111011111111111111111111111<sub>2</sub>  
 ฐานสิบหก = 7 F 7 F F F F F<sub>16</sub>  
 ค่าฐานสิบที่จัดเก็บ (Value actually stored in float)....340282346640×10<sup>38</sup>



## 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 จากระหัส ASCII ของตัวอักษร 0 - 9
2. จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากระหัส 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



## 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 จากระหัส ASCII ของตัวอักษร 0 - 9
2. จงอธิบายวิธีการหาค่าฐานสิบ 0 - 9 จากระหัส 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 จากตาราง ASCII จะได้ว่า
- |          |         |      |
|----------|---------|------|
| ตัวอักษร | เลข '0' | = 48 |
| ตัวอักษร | เลข '1' | = 49 |
| ตัวอักษร | เลข '2' | = 50 |
| ตัวอักษร | เลข '9' | = 57 |

ขนาดฐานสิบจากระยะ ASCII คือ ระหว่าง 48 - 57 ลงด้วย 48

- ② Unicode จะได้ว่า
- |          |         |        |
|----------|---------|--------|
| ตัวอักษร | เลข '0' | = 3664 |
| ตัวอักษร | เลข '๑' | = 3665 |
| ตัวอักษร | เลข '๒' | = 3666 |
| ตัวอักษร | เลข '๓' | = 3673 |

ขนาดฐานสิบจากระยะ ASCII คือ ระหว่าง 3664 - 3673 ลงด้วย 3664

- ③ charset อยู่ View Source เท่ากับ UTF-8 เพราะตัวอักษรไทยมีขนาด โดย  
การเก็บค่าตัวอักษรด้วย UTF-8 อักษรอาจเก็บด้วยขนาด 1-4 Byte โดยตัวอักษรใน  
ตาราง ASCII จะเก็บที่ความยาว 1 Byte ส่วนตัวอักษรอื่นๆ จะเก็บที่ความยาวมากกว่า  
1 Byte