

ภาคผนวก A

การทดลองที่ 1 ข้อมูลและคณิตศาสตร์ในคอมพิวเตอร์

การทดลองนี้เป็นการทบทวนความเข้าใจและแบบฝึกหัดเสริมของเนื้อหาในบทที่ 2 เนื่องจากจำนวนบิตข้อมูลที่ยาวขึ้นจำเป็นต้องใช้โปรแกรมคอมพิวเตอร์ช่วยคำนวณแทน โดยมีวัตถุประสงค์ ดังต่อไปนี้

- เพื่อให้เข้าใจ การแปลง และ คณิตศาสตร์สำหรับ เลขจำนวนเต็มฐานสองชนิดไม่มีเครื่องหมาย และมีเครื่องหมายแบบ 2's Complement
- เพื่อให้เข้าใจการแปลงและคณิตศาสตร์สำหรับเลขทศนิยมฐานสองมาตรฐาน IEEE754 ชนิด Single Precision
- เพื่อให้เข้าใจรหัส ASCII และ Unicode สำหรับข้อมูลตัวอักษร

นอกจากเนื้อหาในบทที่ 2 แล้ว ผู้อ่านสามารถศึกษาเว็บเพจเพิ่มเติม เพื่อทำความเข้าใจอย่างลึกซึ้งได้แก่

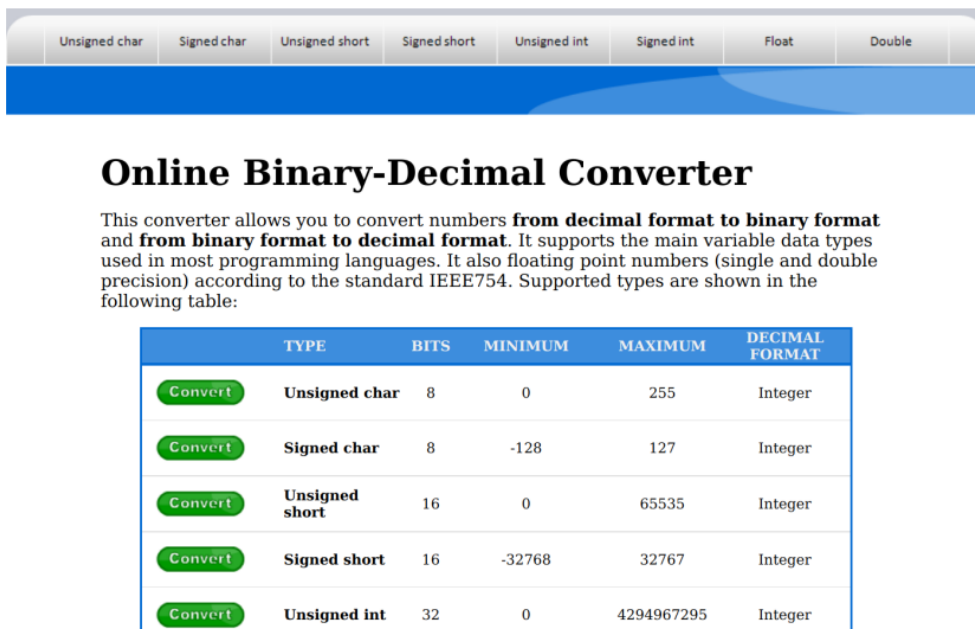
- https://www.tutorialspoint.com/cprogramming/c_data_types.htm
- <https://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html>

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

A.1 การแปลงและคณิตศาสตร์สำหรับเลขจำนวนเต็มฐานสอง

A.1.1 การทดลองแปลงเลขจำนวนเต็มฐานสอง

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



Online Binary-Decimal Converter

This converter allows you to convert numbers **from decimal format to binary format** and **from binary format to decimal format**. It supports the main variable data types used in most programming languages. It also floating point numbers (single and double precision) according to the standard IEEE754. Supported types are shown in the following table:

	TYPE	BITS	MINIMUM	MAXIMUM	DECIMAL FORMAT
Convert	Unsigned char	8	0	255	Integer
Convert	Signed char	8	-128	127	Integer
Convert	Unsigned short	16	0	65535	Integer
Convert	Signed short	16	-32768	32767	Integer
Convert	Unsigned int	32	0	4294967295	Integer

รูปที่ A.1: หน้าเว็บสำหรับแปลงเลขจำนวนเต็มฐานสองเป็นฐานสิบหรือฐานสิบเป็นฐานสองหลายชนิด

- คลิกที่หัวข้อ Signed Char เพื่อทดลองการแปลงเลขจำนวนเต็มมีเครื่องหมายชนิด 2's Complement ขนาด 8 บิต
- 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 1₂
- Hexadecimal (0x) 8 5
- แสดงวิธีทำตามสมการที่ (2.20) ที่ $n=8$ บิตเพื่อแปลงเลขให้ตรงตามรูป

$$\begin{aligned}
 & 10000101 \\
 & = (-1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\
 & = -128 + 4 + 1 \\
 & = -123
 \end{aligned}$$

3. กรอกเลขฐานสองมีเครื่องหมายชนิด 2's Complement 11111111 ขนาด 8 บิต ลงในกล่องข้อความ 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 1₂
- Hexadecimal (0x) F F 8 5
- แสดงวิธีทำตามสมการที่ (2.20) ที่ $n=16$ บิตเพื่อแปลงเลขให้ตรงตามรูป

$$= (-1 \times 2^{15}) + (1 \times 2^{14}) + (1 \times 2^{13}) + (1 \times 2^{12}) + (1 \times 2^{11}) + (1 \times 2^{10}) + (1 \times 2^9)$$

$$(1 \times 2^7) + (1 \times 2^2) + (1 \times 2^0)$$

$$= -32768 + 16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256$$

$$+ 128 + 4 + 1$$

$$= -123$$

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

• Binary $\overset{31}{1} \overset{30}{1} \overset{29}{1} \overset{28}{1} \overset{27}{1} \overset{26}{1} \overset{25}{1} \overset{24}{1} \overset{23}{1} \overset{22}{1} \overset{21}{1} \overset{20}{1} \overset{19}{1} \overset{18}{1} \overset{17}{1} \overset{16}{1} \overset{15}{1} \overset{14}{1} \overset{13}{1} \overset{12}{1} \overset{11}{1} \overset{10}{1} \overset{9}{1} \overset{8}{1} \overset{7}{1} \overset{6}{1} \overset{5}{1} \overset{4}{1} \overset{3}{1} \overset{2}{0} \overset{1}{0} \overset{0}{0} \overset{-1}{1}$

• Hexadecimal (0x) F F F F 8 5

• แสดงวิธีทำตามสมการที่ (2.20) ที่ $n=32$ บิตเพื่อแปลงเลขให้ตรงตามรูป

$$(-1 \times 2^{31}) + (1 \times 2^{30}) + (1 \times 2^{29}) + (1 \times 2^{28}) + (1 \times 2^{27}) + (1 \times 2^{26}) + (1 \times 2^{25}) + (1 \times 2^{24}) + (1 \times 2^{23}) + (1 \times 2^{22}) + (1 \times 2^{21}) + (1 \times 2^{20}) + (1 \times 2^{19}) + (1 \times 2^{18}) + (1 \times 2^{17}) + (1 \times 2^{16}) + (1 \times 2^{15}) + (1 \times 2^{14}) + (1 \times 2^{13}) + (1 \times 2^{12}) + (1 \times 2^{11}) + (1 \times 2^{10}) + (1 \times 2^9) + (1 \times 2^8) + (1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5)$$

A.1.2 คณิตศาสตร์เลขจำนวนเต็มฐานสองขนาด 16 บิต

$= -2147483648 + 1073741824 + 536870912 + 268435456 + 134217728 + 67108864 + 33554432 + 16777216 + 8388608 + 4194304 + 2097152 + 1048576 + 524288 + 262144 + 131072 + 65536 + 32768 + 16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$

$= -123$

ผู้อ่านสามารถศึกษาคณิตศาสตร์เลขจำนวนเต็มฐานสองชนิดไม่มีเครื่องหมาย (unsigned) และชนิดมีเครื่องหมาย (Signed 2's Complement) ไปพร้อมๆ กัน และสอดคล้องกับเนื้อหาในหัวข้อที่ 2.2 โดยผ่านเว็บเบราว์เซอร์ที่ชื่อลิงก์ต่อไปนี้ <http://www.ee.unb.ca/cgi-bin/tervo/alu.pl> ซึ่งเป็น ALU (Arithmetic Logic Unit) Simulator ในวิชา EE3221 Digital Systems II ของภาควิชา Department of Electrical and Computer Engineering มหาวิทยาลัย University of New Brunswick ประเทศ Canada

รูปที่ A.6: หน้าเว็บสำหรับศึกษาการทำงานของวงจรคณิตศาสตร์เลขจำนวนเต็มฐานสอง

ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้

1. แปลงเลขฐานสิบ 32765 และ -32768 เป็นฐานสองชนิด 2's Complement ขนาด 16 บิต ด้วยเว็บแปลงเลขที่ใช้ในการทดลองที่แล้ว

2. ก๊อปปี้เลขฐานสองที่ได้ไปวาง (Paste) จากเว็บแปลงเลขลงในช่อง RB (Register B) และ RA (Register A) ตามลำดับ เลือก ADD แล้วจึงกดปุ่ม Compute เพื่อบวกเลข RB และ RA เข้าด้วยกัน ดังรูป (ไม่ต้องกังวลช่องว่างภายในเลข 16 บิต)

RB
 RA
☒ ADD ☐ SUB ☐ AND ☐ OR ☐ XOR

Result: ADD RC, RB, RA

	RB	7FFD	011111111111101					
	RA	8000	100000000000000	FLAGS				
	RC	FFFD	111111111111101	V C N Z	0	0	1 0	

Integer Interpretation

REGS	UNSIGNED	SIGNED
RB:	(32765)	(+32765)
RA:	+ (32768)	+ (-32768)
RC:	(65533)	(-3)
CHECK:	OK (C=0)	OK (V=0)

รูปที่ A.7: ผลลัพธ์ ADD RC, RB, RA หรือ $RC = RB + RA$ โดย $RB = 011111111111101_2$ และ $RA = 100000000000000_2$ ความยาว 16 บิต

3. อธิบายผลการทดลอง RC (Register C) และ Flag: VCNZ ที่ได้ โดยมองตัวเลข RA, RB และ $RC = RB + RA$ เป็นเลขจำนวนเต็มฐานสองเป็นสองกรณี คือ ฐานสิบหก (คอลัมน์ด้านซ้าย) และ ฐานสอง (คอลัมน์ด้านขวา)

Results: ADD RC, RB, RA หรือ $RC = RB + RA$

RB	7FFD	011111111111101 ₂
RA	8000	100000000000000 ₂
RC	FFFD	111111111111101 ₂

V (Overflow)= 0

C (Carry)= 0 เพราะค่าผลลัพธ์ของการบวกไม่เกิน 16 บิต

N (Negative)= 1 เพราะค่าผลลัพธ์เป็นลบ (MSB คือ 1)

Z (Zero)= 0 เพราะค่าผลลัพธ์ไม่ใช่ศูนย์

4. อธิบายผลการทดลองโดยมองตัวเลข RA, RB และ $RC = RB + RA$ เป็นเลขจำนวนเต็มฐานสองเป็นสองกรณี คือ unsigned (คอลัมน์ด้านซ้าย) และ signed 2's Complement (คอลัมน์ด้านขวา)

Integer Interpretation RB:	32765	+ 32765
RA:	32768	- 32768
RC:	65533	- 3
CHECK:	OK	OK
เพราะเหตุใดจึง OK	เพราะค่า carry = 0	เพราะค่า Overflow = 0

unsigned

signed

A.1.3 คณิตศาสตร์เลขจำนวนเต็มฐานสองขนาด 8 บิต

ผู้อ่านสามารถใช้ความเข้าใจจากการบวกเลขในหัวข้อที่แล้วมาทำการบวกเลขด้วยตนเอง โดยใช้เลขจำนวนเต็มฐานสองขนาด 8 บิตแทน

1. กรอกร เลขที่ได้จาก การ แปลง ฐานสิบ เป็นฐานสองลงในช่องว่างที่จัดไว้ แสดงวิธีทำการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์

		$c_8 \oplus c_7$	c_6	c_5	c_4	c_3	c_2	c_1	c_0	$V = c_8 \oplus c_7$
		$V = \underline{1} \oplus \underline{1} = 0$								
$\times -123$		$\underline{1}$	$\underline{0}$	$\underline{0}$	$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{1}$		0
$+Y \quad + -1$	$+$	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{1}$		
$Z \quad -124$		$\underline{1}$	$\underline{0}$	$\underline{0}$	$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{0}$		0

ซอฟต์แวร์สามารถนำผลลัพธ์ Z ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด

2. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีทำการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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	$V = c_8 \oplus c_7$ $V = \underline{1} \oplus \underline{0}$
		<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	
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>	1
Z		266	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	overflow

ซอฟต์แวร์สามารถนำผลลัพธ์ Z ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด : ไขไม่ได้ เพราะ overflow ช่วง $(-32768, 32767)$

3. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีทำการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 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	$V=c_8 \oplus c_7$ $V=\underline{0} \oplus \underline{0}$
$\times -123$ $+Y \quad +1$	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	0	0
$Z \quad -112$	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>		0

ซอฟต์แวร์สามารถนำผลลัพธ์ Z ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด

4. กรอกเลขที่ได้จากการแปลงลงในช่องว่างที่จัดไว้ แสดงวิธีทำการบวกเลขจำนวนเต็มฐานสองมีเครื่องหมายชนิด 2's Complement ขนาด 8 บิตและคำนวณค่าโอเวอร์โฟลว์ V

[illegible]

ซอฟต์แวร์สามารถนำผลลัพธ์ Z ไปใช้งานต่อได้หรือไม่ เพราะเหตุใด

A.1.4 กิจกรรมท้ายการทดลอง

จงทำการทดลองและตอบคำถามต่อไปนี้ โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.2.2 และตรวจคำตอบตามวิธีทำการทดลองที่ได้ทำไป

1. จงแปลงเลขจำนวนเต็มฐานสิบชนิดไม่มีเครื่องหมายต่อไปนี้ให้เป็นเลขจำนวนเต็มฐานสอง 16 บิต และฐานสิบหกจำนวน 4 หลัก และบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
7	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 ₂	0 x 0 0 0 7 ₁₆
8	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 ₂	0 x 0 0 0 8 ₁₆
15	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 ₂	0 x 0 0 0 F ₁₆
16	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 ₂	0 x 0 0 1 0 ₁₆
255	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ₂	0 x 0 0 F F ₁₆
256	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 ₂	0 x 0 1 0 0 ₁₆
65535	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ₂	0 x F F F F ₁₆
65536	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ₂	0 x F F F F ₁₆ => overflow

2. จง แปลง เลขจำนวน เต็ม ฐาน สิบ ต่อ ไป นี้ ให้ เป็น เลขจำนวน เต็ม ฐาน สอง และ ฐาน สิบ หก ชนิด มี เครื่องหมายแบบ 2's Complement ความยาว 16 บิตแล้วบันทึกผลลัพธ์ที่ได้ลงในตาราง

ฐานสิบ	ฐานสอง	ฐานสิบหก
+1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 ₂	0 x 0 0 0 1 ₁₆
-1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ₂	0 x F F F F ₁₆
+15	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 ₂	0 x F F F 0 ₁₆
-16	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 ₂	0 x F F 0 0 ₁₆
+255	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ₂	0 x 0 0 F F ₁₆
-256	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 ₂	0 x F F 0 0 ₁₆
+65535	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ₂	0 x 7 F F F ₁₆ => overflow
-65536	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ₂	0 x 8 0 0 0 ₁₆ => overflow

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

$$\bullet 1000000000000000 + 0000000000000001$$

$$- \text{ผลลัพธ์} = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1_2$$

$$- \text{ผลลัพธ์} = 8 0 0 1_{16}$$

$$- \text{ผลลัพธ์} = -3 2 7 6 7_{10}$$

$$- \text{โอเวอร์โฟลว์หรือไม่} \dots \text{ไม่ overflow} \dots$$

A.2 การ แปลง และ คณิตศาสตร์ เลข ทศนิยม ฐาน สอง มาตรฐาน IEEE754

การ ทดลอง เพื่อให้ เข้าใจ การ แปลง เลข ทศนิยม ฐานสิบ ให้ เป็น เลข ฐานสอง ตาม รูปแบบ และ ฝึก การ คำนวณ โดยใช้ คณิตศาสตร์มาตรฐาน IEEE754 Single Precision มีความ สอดคล้อง กับ เนื้อหา ใน หัวข้อ ที่ 2.6

A.2.1 เลขทศนิยมชนิดจุดลอยตัวมาตรฐาน IEEE754 Single-Precision

การ ทดลอง นี้ จะ เน้น ที่ การ แปลง เลข ทศนิยม ฐานสิบ ให้ เป็น เลข ทศนิยม ฐานสอง ชนิด จุด ลอยตัว สอดคล้อง กับ เนื้อหา ใน หัวข้อ ที่ 2.6 ใน รูปแบบ Single Precision โดยผ่านเว็บเบราว์เซอร์ที่ผู้อ่านถนัดคลิกที่ชื่อลิงก์ต่อไปนี้

http://www.binaryconvert.com/convert_float.html

เมื่อเว็บเพจปรากฏขึ้น ขอให้ผู้อ่านปฏิบัติตามการทดลอง ดังนี้

1. กรอกเลข 123 ลงในกล่องข้อความ แล้วกดปุ่ม Convert to binary ได้รูปที่ A.8

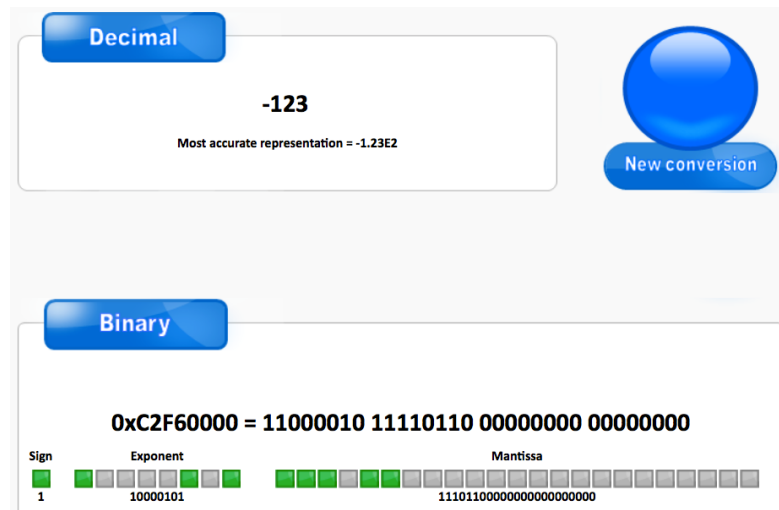
The screenshot shows a web interface for converting decimal numbers to IEEE 754 Single Precision binary. The 'Decimal' section has a text input with '123' and a button 'New conversion'. The 'Binary' section displays the result: **0x42F60000 = 01000010 11110110 00000000 00000000**. Below this, the IEEE 754 fields are visualized: Sign (0), Exponent (10000101), and Mantissa (111011000000000000000000).

รูปที่ A.8: ผลลัพธ์จากการแปลงเลข 123.0 ให้เป็นเลขทศนิยมฐานสองชนิด Single Precision

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

แสดงวิธีทำตามสมการที่ (2.76) เพื่อแปลงเลขให้ตรงตามรูป

2. กรอกเลข -123.0 ลงในกล่องข้อความ แล้วกดปุ่ม Convert to binary ได้รูปที่ A.9



รูปที่ A.9: ผลลัพธ์จากการแปลงเลข -123.0 ให้เป็นเลขทศนิยมฐานสองตามมาตรฐาน IEEE754 ชนิด Single Precision

โปรดสังเกตตำแหน่งของกล่องสี่เหลี่ยมหรือสี่เหลี่ยมที่ตรงกับบิต Sign Exponent และ Mantissa ดังนั้นเราจะเห็นได้ว่าเฉพาะ Sign ที่มีการเปลี่ยนแปลง

แสดงวิธีทำตามสมการที่ (2.76) เพื่อแปลงเลขให้ตรงตามรูป

3. คลิกบนลิงก์นี้ เพื่อทดลองบวกและคูณเลขในรูปแบบ Single Precision ด้วยลิงก์ต่อไปนี้ <http://weitz.de/ieee/> เลื่อนหน้าเว็บลงไปที่ด้านล่างสุด เพื่อค้นหาแถบเมนูตามรูปที่ A.10 แล้วกดเลือกปุ่ม binary32 เพื่อทดลองการบวกและคูณเลข IEEE754 Single Precision

binary16 binary32 binary64 binary128

รูปที่ A.10: 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;"> + - × / </div>			
0.0	0 +	0 . 000000000000000000000000 0.0 0x00000000 0b00000000000000000000000000000000	00000000 +0

รูปที่ A.11: ผลลัพธ์จากการบวกเลข -123.0+123.0 ให้เป็นเลขทศนิยมฐานสองชนิด IEEE754 Single Precision

จะสังเกตเห็นว่า ผลลัพธ์ที่ได้เรียกว่า True Zero ตามตารางที่ 2.12

5. กดปุ่ม \times (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 style="display: flex; justify-content: space-around; margin-bottom: 10px;"> + - × / </div>			
-15129.0	1 -	1 . 110110001100100000000000 1.8468018 0xC66C6400 0b11000110011011000110010000000000	10001100 +13

รูปที่ A.12: ผลลัพธ์จากการคูณเลข -123.0 \times 123.0 ให้เป็นเลขทศนิยมฐานสองชนิด IEEE754 Single Precision

แสดงวิธีทำตามสมการที่ (2.76) เพื่อแปลงเลขให้ตรงตามผลคูณในรูปที่ A.12

A.2.2 กิจกรรมท้ายการทดลอง

จงใช้ลิงก์ของเว็บเพจต่อไปนี้ในการตอบคำถาม

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

[illegible]

รูปที่ A.13: เว็บสำหรับการตอบคำถามเพื่อสร้างเลขหรือแปลงเลขฐานสิบด้วยมาตรฐาน IEEE754 Single Precision การกดเลือกคือทำให้บิตนั้นเท่ากับ '1'

โดยแสดงวิธีทำตามเนื้อหาในหัวข้อที่ 2.6 และตรวจคำตอบตามวิธีทำการทดลองที่ได้ทำไป 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 0₂

ฐานสิบหก = 8 0 0 0 0 0 0 0₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....0

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

2. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ -1.0_{10} โดยการกดเลือกปุ่มสี่เหลี่ยมในส่วน Exponent เท่านั้น ต่อจากข้อที่แล้ว

เลขฐานสอง = 1011 | 1111 | 1000 | 0000 | 0000 | 0000 | 0000 | 0000₂

จำนวนสิบหก = $B F 8 0 0 0 0 0_{16}$

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 = 1011 | 1111 | 1100 | 0110 | 0110 | 0110 | 0110 | 0110₂

ฐานสิบหก = $\underline{B} \quad \underline{F} \quad \underline{C} \quad \underline{b} \quad \underline{b} \quad \underline{b} \quad \underline{b} \quad \underline{b}_{16}$

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....-1.5499...

ความผิดพลาด (Error due to conversion).....4.76837...

4. จง สร้าง เลข ทศนิยม ฐาน สอง IEEE754 ที่มี ค่า เท่ากับ $1.17549435082 \times 10^{-38}$ หรือ $1.17549435082e-38$ ซึ่งเป็นค่า normalized ที่น้อยที่สุด (Normalize)

เลขฐานสอง = 0000 0000 1 000 0000 0000 0000 0000 0000

ฐานสิบหก = 0 0 8 0 0 0 0 0₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float) 1.1754943508222875

ความผิดพลาด (Error due to conversion) 2.2875079

5. จง สร้าง เลข ทศนิยม ฐาน สอง IEEE754 ที่มี ค่า เท่ากับ $1.17549421069 \times 10^{-38}$ หรือ $1.17549421069e-38$ ซึ่ง อยู่ใน รูป ดี นอร์มัลไลซ์ (Denormalize) เพราะ มี ค่า น้อย กว่า ค่า นอร์มัลไลซ์ที่ต่ำที่สุด

เลขฐานสอง = 0000|0000|๑111|1111|1111|1111|1111|1111₂

ฐานสิบหก = 0 0 7 F F F F F₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....1.175494210692

ความผิดพลาด (Error due to conversion). 2.44107548702944

6. จง สร้าง เลข ทศนิยม ฐาน สอง IEEE754 ที่มี ค่า เท่ากับ $1.40129846432 \times 10^{-45}$ หรือ $1.40129846432e-45$ ซึ่งอยู่ในรูป ดินอร์มัลไลซ์ (Denormalize) และต่ำที่สุด

เลขฐานสอง = 0000 0000 0000 0000 0000 0000 0000 0001₂

ฐานสิบหก = 0 0 0 0 0 0 0 1₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float) 1.401298464 324817

ความผิดพลาด (Error due to conversion) 4.817070923729

7. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ 1.0×10^{-46} หรือ $1e-46$ ซึ่งอยู่ในรูป ดินออร์ม

ลไคลซ์ (Denormalize) และจัดเก็บด้วยค่า 0.0 แทน

เลขฐานสอง = 00000000000000000000000000000000₂

ฐานสิบหก = 0 0 0 0 0 0 0 0₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float).....0

ความผิดพลาด (Error due to conversion)..... -1×10^{-46}

8. จง สร้าง เลข ทศนิยม ฐาน สอง IEEE754 ที่มี ค่า เท่ากับ $3.40282346640 \times 10^{38}$ หรือ $3.40282346640e38$ ซึ่งเป็นค่านอร์มัลไลซ์ที่มากที่สุด

เลขฐานสอง = 01111111011111111111111111111111₂

ฐานสิบหก = 7 F 7 F F F F F₁₆

ค่าฐานสิบที่จัดเก็บ (Value actually stored in float)..... $34028234663852885 \times 10^{38}$

ความผิดพลาด (Error due to conversion)..... $-14711401983 \times 10^{27}$

9. จงสร้างเลขทศนิยมฐานสอง IEEE754 ที่มีค่าเท่ากับ 3.5×10^{38} หรือ $3.5e+38$ ซึ่งมากกว่าค่านอร์มัลไลซ์ที่มากที่สุด ซึ่งหมายถึงค่าอนันต์ (∞ : Infinity) ตามตารางที่ 2.12

เลขฐานสอง = 01111111100000000000000000000000₂

ฐานสิบหก = 7 F 8 0 0 0 0 0₁₆

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

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

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

เลขฐานสอง = 01111111111111111111111111111111₂

ฐานสิบหก = 7 F F F F F F F₁₆

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

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

A.3 รหัสของข้อมูลตัวอักษร

A.3.1 การทดลอง

การทดลองในหัวข้อนี้ จะเป็นการ แปลง รหัส ตัว อักษร ภาษา อังกฤษ และ ไทย เป็น รหัส ASCII และ Unicode ชนิด UCS-2 ตาม เนื้อหา ใน หัวข้อ 2.7 ผ่าน ทาง เว็บไซต์ <https://www.branah.com/ascii-converter> ที่มีนักพัฒนาเพื่อเผยแพร่ความรู้เป็นวิทยาทานเช่นเดียวกับเว็บที่ได้ทดลองมา

1. เปิดเว็บตามลิงก์ต่อไปนี้ หรือ กดปุ่มซ้ายบนชื่อลิงก์
<https://www.branah.com/ascii-converter>

2. กรอกข้อความต่อไปนี้ ลงในกล่องข้อความ 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

ASCII (Example: a b c)

ไ ท ย ก ข ค a b c

☐ Convert white space characters

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

e44 e17 e22 e01 e02 e04 61 62 63

Decimal (Example: 97 98 99)

3652 3607 3618 3585 3586 3588 097 098 099

Binary (Example: 01100001 01100010 01100011)

111001000100 111000010111 111000100010 111000000001 111000000010 111000000100 01100001 01100010 01100011

Base64 (Example: YSBiIGM=)

RCAXIClgASACIAQgYSBiIGM=

รูปที่ A.14: ผลลัพธ์จากการกรอกและแปลงตัวอักษร ไ ท ย ก ข ค a b c เป็นรหัสต่างๆ

- ### A.3.2 กิจกรรมท้ายการทดลอง

- ③. charset ของ View Source เท่ากับ UTF-8 เพราะตัวอักษรไทยไม่แสดงออก
โดยการเก็บตัวอักษรด้วย UTF-8 ตัวละครเก็บตัวความยาว 1-4 Byte โดยตัวอักษรใน
กรอบ ASCII จะเก็บที่ความยาว 1 Byte ส่วนตัวอักษรภาษาอื่น ๆ จะเก็บความยาวมากกว่า
1 Byte