



กระบวนวิชา 204203

Lab	
HW	

ปฏิบัติการ 2

Data Representation – Part II

ข้อกำหนด

- ในข้อที่ระบุว่ามี **[Attachments]** ให้ Download ไฟล์ Template จาก Grader ลงมาแล้วส่งเฉพาะไฟล์ที่ชื่อตรงกับระบุในแต่ละข้อเท่านั้น
- ข้ออื่น ๆ ให้เขียน main function เองแยกไว้อีกไฟล์ในลักษณะเดียวกันกับการบ้านในสัปดาห์แรกแล้วส่งเฉพาะไฟล์ที่ implement ฟังก์ชันที่ระบุ (ไม่ Upload ไฟล์ที่มี main function)

1) 4 คะแนน (Lab02_1_XXXXXXXXX.go) ให้เขียนฟังก์ชัน Boolean `powerOfTwo(x uint64) bool`

เพื่อตรวจสอบว่าจำนวนเต็มบวก n สามารถเขียนในรูปของ 2^x ($0 \leq x < 64$) เมื่อ x เป็นจำนวนเต็ม ($0 \leq x$)

ได้หรือไม่ โดยใช้ bitwise operation ในการตรวจสอบโดยพิจารณาจากคุณสมบัติที่ว่าหาก n อยู่ในรูปของ 2^x แล้ว $n - 1$ & n จะเท่ากับ 0 เช่น

$$\begin{array}{lll}
 n & = 16 & = 10000_2 \\
 n - 1 & = 15 & = 01111_2 \\
 n \& (n - 1) & & = 00000_2
 \end{array}$$

ทั้งนี้ไม่อนุญาตให้ใช้ operation คุณหาร modulo (*, /, %), หรือฟังก์ชันมาตรฐานจาก library **math** ในการแก้ปัญหา

Input	Output
18	0
1024	1

2) 4 คะแนน (Lab02_2_XXXXXXXXX.go) **[Attachments]** ให้แก้ไขฟังก์ชัน `addition()` จาก Lab01_1 ในสัปดาห์ก่อนให้รองรับจำนวนจริงบวก $r1$ และ $r2$ ที่มีความยาวไม่เกิน 70 หลัก (รวมจุดทศนิยมแล้ว) ในฐาน n ($2 \leq n \leq 10$) โดย function จะมี signature ดังนี้ `baseNAddition(r1, r2 string, n int) string`

Input	Output
11.01 1.1 2	100.11
18.50 10.2 10	28.70

- 3) **6 คะแนน** (HW02_1_XXXXXXXXX.go) **[Attachments]** ให้เขียนโปรแกรม เพื่อแปลงระหว่าง unsigned 32 บิต และ string แทน IP address version 4 (IPv4) เลข IP address v4 มีทั้งหมด 32 บิตหรือ 4 ไบต์ แต่ละไบต์จะมีค่าระหว่าง 0–255 และถูกคั่นด้วยจุด (.) เช่น 10.4.28.0 โดยมีรายละเอียดดังนี้
- ไม่อนุญาตให้ใช้ operation คูณ หหาร modulo (*, /, %), หรือฟังก์ชันมาตรฐานจาก library **math**, **net**, หรือ **encoding/binary** ในการแก้ปัญหา
 - ให้เขียนฟังก์ชัน `ipv4Decode(ipUint uint32) string` เพื่อแปลงและคืนค่า string ของหมายเลข IP โดยคำนวณจาก จำนวนเต็ม 32 บิต แบบ unsigned `ipUint`
 - ให้เขียนฟังก์ชัน `ipv4Encode(ipString string) uint32` เพื่อแปลงและคืนค่าเป็นจำนวนเต็ม 32 บิต แบบ unsigned โดยคำนวณจาก string `ipString` ที่แทน IP address v4
 - โปรแกรมจะต้องอ่าน input โดยวิธี redirection จากไฟล์ โดย input ไฟล์จะมีลักษณะดังนี้
 - บรรทัดที่ 1 จะเป็นจำนวน n ($1 \leq n \leq 2000$) แทน จำนวน test case ทั้งหมดในไฟล์
 - บรรทัดถัดไป n บรรทัด จะเป็น test case
 - ในแต่ละบรรทัดจะเป็น operation ที่ต้องทำ โดย
 - หากเป็น string ที่คั่นด้วยจุด (.) จะเป็นการแปลงจาก IP Address เป็นจำนวนเต็มแบบ unsigned
 - หากเป็นจำนวนเต็มปกติจะเป็นการแปลงจากจำนวนเต็มแบบ unsigned เป็น IP Address

ตัวอย่างไฟล์ input.txt

ตัวอย่าง output

2	192.168.6.13
3232237069	169608194
10.28.4.2	

Hint:

- นักศึกษาสามารถศึกษาความสัมพันธ์ระหว่างตัวเลข 32-bit binary, unsigned decimal และหมายเลข IP ได้จาก <https://goo.gl/MvxaaS> และตรวจสอบการเข้ารหัสข้อมูล IP address ได้จาก <https://goo.gl/7RRtqP>
- พิจารณาการใช้ function `Itoa()` และ `Atoi()` จาก library **strconv**

- 4) 6 คะแนน (HW02_2_XXXXXXXXX.go) ให้เขียนให้เขียนฟังก์ชัน `roundToEven(x string, bPlace uint8)` เพื่อคืนค่า binary string แทนจำนวน x ที่ผ่านการปัดเศษแบบ round to even ให้เหลือจำนวนตำแหน่ง binary places (จำนวนหลักหลังจุดทวินิยม) ตามที่ระบุจากตัวแปร `bPlace`

<u>Input</u>	<u>Output</u>
1101.101 2	1101.10
1101.101 0	1110
1101.111 2	1110.00
1101.111 5	1101.11100

- 5) คะแนน (HW02_3_XXXXXXXXX.go) ให้เขียนฟังก์ชัน `float16bitNormed(n float32) string` เพื่อคืนค่า binary string แทนการแทนค่าข้อมูลแบบ float ความยาว 16 บิตตามวิธีของ IEEE 754 แบบ normalized (Case 1 จาก slide เรื่อง Data Representation Part II) โดยให้มีความยาวบิตในแต่ละส่วนดังรูปด้านล่าง ทั้งนี้ให้ใช้การปัดเศษแบบ **truncate (ตัดทิ้งตามความยาวที่ระบุโดยไม่ต้องปัด)** ในส่วน frac และหากมีการ overflow ให้คืนค่า empty string

Hint: พิจารณาเรียกใช้ฟังก์ชัน `floatToBaseB()` จาก Lab02_1 ในสัปดาห์ก่อน และ ฟังก์ชันต่าง ๆ จาก library strings



Input	Output (แสดงผลแบบเว้นวรรคให้อ่านง่าย)
204203	0 10010000 1000111
0.002	0 01110110 0000011
-23	1 10000011 0111000
0.00000000000000000000000000000001	0 00001110 0000100
0.000000000000000000000000000000005	0 00000011 0001000
0.000000000000000000000000000000002	0 00000001 1000000
0.0000000000000000000000000000000012	0 00000001 0000000
0.000000000000000000000000000000000001	
338953138925153547590470800371487866880	0 11111110 1111111

การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน canvas รายวิชา
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยัง website ที่ใช้ส่งการบ้าน <http://cmu.to/gdr203>