

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....
กำหนดส่ง



กระบวนวิชา **204203**

Lab	
HW	
Until	

การบ้านปฏิบัติการ 4

Data Representation III (20 คะแนน)

ข้อกำหนด

- ในข้อที่มี **[Attachment]** ให้ Download ไฟล์ Template จาก Grader ลงมาแล้วส่งเฉพาะไฟล์ที่ชื่อตรงกับระบุในแต่ละข้อเท่านั้น และสามารถสร้างฟังก์ชันย่อยเพิ่มเติมได้ตามอัธยาศัย
- Binary string ในปัญหาทุกข้อ หากเป็น integer จะใช้การแทนข้อมูลแบบ two's complement

- 1) 4 คะแนน (HW04_1_XXXXXXXXX.go) ให้เขียนฟังก์ชัน Boolean `powerOfTwo(n 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 คุณ หาร modulo (*, /, %), หรือฟังก์ชันมาตรฐานจาก library **math** ในการแก้ปัญหา

Input	Output
18	false
1024	True

- 2) 4 คะแนน (HW04_2_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 ด้วย **Bitwise Operation** โดยมีรายละเอียดดังนี้
- ไม่อนุญาตให้ใช้ operation คุณ หาร 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

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

- บรรทัดถัดไป n บรรทัด จะเป็น test case ในแต่ละบรรทัดจะเป็น operation ที่ต้องทำ โดย
 - o หากเป็น string ที่คั่นด้วยจุด (.) จะเป็นการแปลงจาก IP Address เป็นจำนวนเต็มแบบ unsigned
 - o หากเป็นจำนวนเต็มปกติจะเป็นการแปลงจากจำนวนเต็มแบบ unsigned เป็น IP Address

ตัวอย่างไฟล์ input.txt

ตัวอย่าง output

2 3232237069 10.28.4.2	192.168.6.13 169608194
------------------------------	---------------------------

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

- 3) 4 คะแนน (HW04_3_XXXXXXXXX.go) [Attachments] ให้เขียนฟังก์ชัน decodeHuffman(encodedText string, codingTable map[string]string) string เพื่อทำการถอดรหัสข้อมูลที่ผ่านการบีบอัดด้วยวิธีบีบอัดข้อมูลแบบ Huffman Coding โดยรับข้อมูล Input เป็นข้อมูลที่ผ่านการเข้ารหัสแล้ว encodedText และตาราง codingTable ของข้อความนั้น ในรูปแบบ Map

ตัวอย่าง output เมื่อทำการเรียกใช้โปรแกรมด้วย Command Redirection จากฟังก์ชัน main() ที่มีให้เช่น

```
$ go build -o main *.go ; ./main < input.txt
```

input.txt

ตัวอย่าง output

{000: , 001:r, 010:H, 011:W, 10:l, 110:o, 1110:e, 1111:d} 01011101010110000011110001101111	Hello World
{0000:L, 0001:u, 0010:d, 00110:a, 00111:., 010:o, 011:m, 100: , 1010:r, 10110:p, 10111:l, 1100:s, 1101:t, 1110:i, 1111:e} 0000010101011110111001110101101000 0101110000100101011101010100110011 10110110000110011111110100111	Lorem ipsum dolor sit amet.

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

- 4) 4 คะแนน (HW04_4_XXXXXXX.go) [Attachments] ให้เขียนฟังก์ชัน decodeLZ77(matches []Triplet) string เพื่อทำการถอดรหัสข้อมูลผ่านการบีบอัดด้วยวิธีบีบอัดข้อมูลแบบ LZ77 และในรูปของ Slice ของชนิดข้อมูล Triplet เมื่อกำหนดนิยามชนิดข้อมูล Triplet ดังนี้

```
type Triplet struct {
    Offset int
    Length int
    Next byte
}
```

ทั้งนี้ฟังก์ชัน จะต้องไม่นำ ASCII code สำหรับตัวอักษร NUL (0) มาแสดงผล

ตัวอย่าง output เมื่อทำการเรียกใช้โปรแกรมด้วย Command Redirection จากฟังก์ชัน main() ที่มีให้เช่น

```
$ go build -o main *.go ; ./main < input.txt
```

input.txt

ตัวอย่าง output

(0,0,97), (0,0,98), (2,2,99), (4,3,97), (2,2,97), (0,0,10)	ababcbababaa
(0,0,97), (0,0,98), (0,0,114), (3,1,99), (2,1,100), (7,4,10)	abracadabra
(0,0,104), (0,0,97), (0,0,110), (1,1,97), (5,1,32), (0,0,98), (7,2,97), (2,2,10)	hannah banana
(0,0,99), (0,0,97), (0,0,116), (0,0,10), (4,4,0)	cat cat

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

- 5) 4 คะแนน (HW04_5_XXXXXXX.go) [Attachments] ให้เขียนฟังก์ชัน `encodeLZ77(windowSize int, bufferSize int, inputString string) []Triplet` เพื่อทำการเข้ารหัส `inputString` ด้วยวิธีบีบอัดข้อมูลแบบ LZ77 เมื่อกำหนดขนาดของ search window ด้วยตัวแปร `windowSize` และ กำหนดขนาดของ look ahead buffer ด้วยตัวแปร `bufferSize` ทั้งนี้ฟังก์ชันจะคืนค่าเป็น slice ของชนิดข้อมูล Triplet เมื่อกำหนดนิยามชนิดข้อมูล Triplet ดังนี้

```
type Triplet struct {
    Offset int
    Length int
    Next byte
}
```

ทั้งนี้ฟังก์ชันจะต้อง Encode การไม่มีตัวอักขระเพิ่มใน Triplet ด้วยตัวอักษร NUL (0) (ดูจากตัวอย่างบรรทัดล่างสุด)

ตัวอย่าง output เมื่อทำการเรียกใช้โปรแกรมด้วย Command Redirection จากฟังก์ชัน `main()` ที่มีให้

<u>input.txt</u>	<u>ตัวอย่าง output</u>
4 4 ababcbababaa	(0,0,97), (0,0,98), (2,2,99), (4,3,97), (2,2,97), (0,0,10)
2 2 ABCAABC	(0,0,65), (0,0,66), (0,0,67), (0,0,65), (1,1,66), (0,0,67), (0,0,10)
3 3 ABCAABC	(0,0,65), (0,0,66), (0,0,67), (3,1,65), (0,0,66), (0,0,67), (0,0,10)
5 5 cat cat	(0,0,99), (0,0,97), (0,0,116), (0,0,10), (4,4,0)

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 ไปยังระบบตรวจให้คะแนนอัตโนมัติ <https://cmu.to/gdr203>

COMPUTER SCIENCE
Chiang Mai University