

01 กำหนดโครงสร้างข้อมูล enum

Recent submission on 14:47 04 Mar 2020 Result: PASSED [PPPPPPPP]

01 - กำหนดโครงสร้างข้อมูล enum

02 - คำนวณวันในสัปดาห์

03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง

04 - Word Count

05 - Single Linked List

06 - จำนวนบิตที่ไม่ใช่ 0

07 - File Average Score

08 - Stock Log File

กำหนดโครงสร้างข้อมูล enum

ให้กำหนดโครงสร้างข้อมูล enum ที่มีชื่อว่า DaysOfWeek เพื่อใช้เก็บค่าวันในสัปดาห์ตั้งแต่วันจันทร์ถึงวันอาทิตย์ โดยให้ใช้ตัวอักษรย่อว่า MON, TUE, WED, THU, FRI, SAT, และ SUN ตามลำดับ นอกจากนี้ ให้มีการกำหนดลำดับของวันดังต่อไปนี้

- วันที่ถัดจากวันอาทิตย์ออกไป 1 วันคือวันจันทร์ และวันที่ถัดจากวันจันทร์ออกไปอีก 1 วันคือวันอังคาร และถัดแบบนี้ไปเรื่อยๆ จนถึงวันที่ถัดจากวันศุกร์ออกไปอีก 1 วันคือวันเสาร์ (ส่วนวันที่ถัดจากวันเสาร์ออกไปนั้น นักเรียนไม่ต้องสนใจ)

ให้กำหนดโครงสร้างนี้ในบริเวณที่เว้นไว้ให้ เพื่อให้โปรแกรมทำงานถูกต้อง ตัวอย่างของผลลัพธ์ได้แสดงไว้ข้างล่างนี้

ตัวอย่างผลลัพธ์

```
The day after Monday is Tuesday
Two days after Monday is Wednesday
```

```
#include <stdio.h>
#include <string.h>
```

[hide line #]

```
int main()
{ char day_names[][10]={ " ", " ", " ",
                          "Sunday", "Monday", "Tuesday",
                          "Wednesday", "Thursday", "Friday",
                          "Saturday"};

enum DaysOfWeek today;
```

```
today = MON;
printf("The day after %s is %s\n", day_names[today], day_names[today+1]);
printf("Two days after %s is %s\n", day_names[today], day_names[today+2]);
}
```

[hide line #]

01 - กำหนดโครงสร้างข้อมูล enum ✓02 - คำนวณวันในสัปดาห์ ✓03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง ✓04 - Word Count ✓05 - Single Linked List ✗06 - จำนวนบิตที่ไม่ใช่ 0 ✓07 - File Average Score ✓08 - Stock Log File ✗

Current Recent All

หาวันในสัปดาห์

ให้นิสิตกำหนด enum ของ DayOfWeek และเขียนฟังก์ชันชื่อ findDayOfWeek() เพื่อหาวันของสัปดาห์จากวันและเดือน (เป็นตัวเลข) ของปีพ.ศ. 2556 ที่ผู้ใช้พิมพ์เข้ามา สังเกตว่าวันที่ 1 เดือน 1 ของปีพ.ศ. 2556 นั้นเป็นวันอังคาร

ตัวอย่าง

```
Day:1 Month:1 of Year 2013 is Tuesday.
Enter Day and Month: 26 7
Day:26 Month:7 of Year 2013 is Friday.
```

[hide line #]

#include <stdio.h>

DayOfWeek findDayOfWeek(int day, int month) {

}

```
int main()
{
```

```
    int day, month;
    DayOfWeek dow;
```

```
    printf("Day:1 Month:1 of Year 2013 is Tuesday.\n");
    printf("Enter Day and Month: ");
    scanf("%d %d", &day, &month);
    dow = findDayOfWeek(day, month);
    printf("Day:%d Month:%d of Year 2013 is ", day, month);
    switch(dow){
```

```
        case : printf("Sunday.\n"); break;
```

```
        case : printf("Monday.\n"); break;
```

```
        case : printf("Tuesday.\n"); break;
```

```
        case : printf("Wednesday.\n"); break;
```

```
        case : printf("Thursday.\n"); break;
```

```
        case : printf("Friday.\n"); break;
```

```
        case : printf("Saturday.\n"); break;
```

```
    }
```

```
}
```

Current

Recent

All

01 - กำหนดโครงสร้างข้อมูล enum

02 - คำนวณวันในสัปดาห์

03 - Struct ที่เก็บพิกัดของจุด และหาระยะทาง

04 - Word Count

05 - Single Linked List

06 - จำนวนบิตที่ไม่ใช่ 0

07 - File Average Score

08 - Stock Log File

จงเขียนโปรแกรมเพื่อรับค่าพิกัดจุดเริ่มต้น และจุดสิ้นสุดบนแกนสองมิติ (X, Y) จำนวน n คู่จากผู้ใช้งาน แล้วคำนวณและแสดงผลระยะห่างระหว่างจุด 2 จุดอย่างต่อเนื่องตั้งแต่จุดที่ 1 ถึงจุดที่ N

โปรแกรมมีโครงสร้างของชุดคำสั่งดังนี้

1. ประกาศโครงสร้างข้อมูล struct Point เพื่อใช้เก็บตำแหน่งของจุดทั้ง 2 ที่อยู่บนแกน X, Y ซึ่งมีชนิดข้อมูลเป็น double
2. ที่ main() ประกาศตัวแปร array of struct ชื่อ points ให้มีชนิดข้อมูลเป็น struct Point เพื่อเก็บคู่อันดับของพิกัด X, Y จากผู้ใช้ จำนวน n คู่
3. เขียนคำสั่ง for เพื่ออ่านคู่อันดับของพิกัด X, Y จำนวน n คู่จากผู้ใช้งาน ตามตัวอย่างการรับค่าจาก standard input
4. เขียนคำสั่ง for โดยกำหนดให้ภายใน for loop ประกอบด้วย statements ต่อไปนี้
 - o กำหนดตัวแปร dx, dy ให้มีชนิดข้อมูลเป็น double ใช้เก็บความยาวจากจุดหนึ่งไปยังอีกจุดหนึ่งตามแนวแกน X และแนวแกน Y ตามลำดับ
 - o คำนวณระยะห่าง (length) ระหว่างจุด 2 จุด
 - o แสดงผลลัพธ์ตามรูปแบบที่กำหนดไว้ในตัวอย่างการแสดงผล (ต้องการทศนิยม 2 ตำแหน่ง)

ตัวอย่าง โปรแกรม

```
Number of Points: 5
P1.X: 1
P1.Y: 1
P2.X: 1
P2.Y: -5
P3.X: -5
P3.Y: 25
P4.X: 19
P4.Y: -3
P5.X: 15
P5.Y: 20
Length:
Length from P1(1.00, 1.00) to P2(1.00, -5.00) is 6.00
Length from P2(1.00, -5.00) to P3(-5.00, 25.00) is 30.59
Length from P3(-5.00, 25.00) to P4(19.00, -3.00) is 36.88
Length from P4(19.00, -3.00) to P5(15.00, 20.00) is 23.35
```

- 01 - กำหนดโครงสร้างข้อมูล enum
- 02 - คำนวณวันในสัปดาห์
- 03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง
- 04 - Word Count
- 05 - Single Linked List
- 06 - จำนวนบิตที่ไม่ใช่ 0
- 07 - File Average Score
- 08 - Stock Log File

04 Word Count

Recent submission on 18:29 07 Mar 2020 Result: PASSED [PPPPP]

CurrentRecentAll

Word Count

จงเขียนโปรแกรมในการนับจำนวนคำที่ได้รับมาจากผู้ใช้งาน โดยผู้ใช้งานจะป้อนคำเข้ามาในระบบเรื่อย ๆ โปรแกรมสามารถบันทึกข้อมูลจำนวนคำที่ไม่ซ้ำกันได้ไม่เกิน 20 คำ (array data) และแต่ละคำ (word) มีความยาวไม่เกิน 20 ตัวอักษร

เมื่อผู้ใช้งานป้อนคำเข้าไปในระบบจนพอใจแล้วให้ หยุดโปรแกรมโดยการพิมพ์คำว่า exit

โปรแกรมจะหยุดทำงานพร้อมกับพิมพ์คำ (word) และจำนวนคำ (count) ที่ได้รับเข้ามาในระบบ

ตัวอย่าง 1

This
Is
BOOK
Book
Hello
Hello
Hello
exit

Output:
This = 1
Is = 1
BOOK = 1
Book = 1
Hello = 3

ตัวอย่าง 2

one
one
one
exit

Output:
one = 3

[hide line #]

```
#include<stdio.h>
#include<string.h>

struct Word {
    [redacted]

};

int main(){
    char input[21];
    int i, last_word = 0, find = 0;
    struct Word data[20] = {0};
    scanf("%20s", input);

    while([redacted]){

        [redacted]

    }

    printf("Output:\n");
    for(i=0;i<last_word;i++){
        printf("%s = %d\n", data[i].word, data[i].count);
    }

}
```

- 01 - กำหนดโครงสร้างข้อมูล enum
- 02 - คำนวณวันในสัปดาห์
- 03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง
- 04 - Word Count
- 05 - Single Linked List
- 06 - จำนวนบิตที่ไม่ใช่ 0
- 07 - File Average Score
- 08 - Stock Log File

05 Single Linked List

Recent submission on 23:14 12 Mar 2020 Result: FAILED [-----]

Current

Recent

All

สร้าง list อย่างง่าย

ให้นักเขียนโปรแกรมเพื่อสร้างลิสต์อย่างง่ายใช้เองโดยไม่ใช้ array แต่ละ element ในลิสต์จะเป็น node (struct) ที่มีตัวแปร value (int) เป็นค่าที่เก็บไว้ใน node และตัวแปร next ซึ่งจะชี้ไปยัง node ต่อไป ค่า next ของ node สุดท้ายจะเป็น NULL เราสามารถเรียกใช้ลิสต์ผ่าน pointer ซึ่งชี้ไปยัง โหนดแรกของลิสต์นั้น

ฟังก์ชันที่ใช้กับลิสต์มีสองฟังก์ชันคือ insertNode() ซึ่งจะเพิ่ม node เข้าไปในลิสต์ โดยจะรับค่าที่จะเพิ่มและ pointer ซึ่งชี้ไปที่โหนดแรกของลิสต์ แล้วจะทำการสร้าง โหนดใหม่ที่เก็บค่านั้นและเพิ่มเข้าไปในลิสต์ การเพิ่มโหนดนั้นจะเรียงลำดับค่าของ โหนดจากน้อยไปมากให้ด้วย (ดังนั้นลิสต์จะเรียงลำดับค่าอยู่เสมอ) ฟังก์ชันที่สองคือ printList() ซึ่งจะพิมพ์ค่าของ โหนดในลิสต์ออกมาทีละค่าโดยเว้นช่องว่างระหว่างค่าและขึ้นบรรทัดใหม่เมื่อพิมพ์ค่าสุดท้าย

คำแนะนำ

- การเพิ่มโหนดนั้นอาจเพิ่มได้ทั้งท้ายลิสต์ หน้าลิสต์ และระหว่างกลาง ควรเขียนโค้ดให้ครบทุกกรณี
- ระวังกรณีพิเศษคือเมื่อลิสต์ยังว่างอยู่ตอนใส่โหนดครั้งแรก

[hide line #]

```
#include <stdio.h>
#include <stdlib.h>

[ ]

void printList(node *pList) {
    [ ]
}

[ ] insertNode( [ ] ) {
    [ ]

}

int main() {
    int i, value;
    node *pList=NULL;

    for (i = 0; i < 10; i++) {
        scanf(" %d", &value);
        [ ]
    }

    printList(pList);
}
```

01 - กำหนดโครงสร้างข้อมูล enum ✓

02 - คำนวณวันในสัปดาห์ ✓

03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง ✓

04 - Word Count ✓

05 - Single Linked List ✗

06 - จำนวนบิตที่ไม่ใช่ 0 ✓

07 - File Average Score ✓

08 - Stock Log File ✗

06 จำนวนบิตที่ไม่ใช่ 0

Recent submission on 16:29 04 Mar 2020 Result: PASSED [PPPPP]

Current

Recent

All

จำนวนบิตที่ไม่ใช่ 0

จงเติม โปรแกรมที่รับค่าจำนวนเต็มที่อยู่ในช่วงที่ตัวแปรประเภท unsigned char (8 บิต) สามารถเก็บข้อมูลได้ แล้วพิมพ์จำนวนบิตที่ไม่เป็น 0 ของค่านั้นออกมาดังตัวอย่าง

หมายเหตุ ให้ใช้ bitwise operators และ arithmetic operators ในการนับจำนวนบิตที่ไม่เป็น 0 เท่านั้น

ตัวอย่างผลลัพธ์ที่ 1

```
Enter an 8-bit number: 255
The number 255 has 8 non-zero bits
```

ตัวอย่างผลลัพธ์ที่ 2

```
Enter an 8-bit number: 7
The number 7 has 3 non-zero bits
```

```
#include <stdio.h>
```

[hide line #]

```
int main()
{
    unsigned char x;
    int n;
```

```
    printf("Enter an 8-bit number: ");
    scanf("%d", &n);
    x = (unsigned char)n;
    int numbits =
```

```
    printf("The number %d has %d non-zero bits\n", x, numbits);
    return 0;
}
```

- 01 - กำหนดโครงสร้างข้อมูล enum ✓
- 02 - คำนวณวันในสัปดาห์ ✓
- 03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง ✓
- 04 - Word Count ✓
- 05 - Single Linked List ✗
- 06 - จำนวนบิตที่ไม่ใช่ 0 ✓
- 07 - File Average Score ✓
- 08 - Stock Log File ✗

07 File Average Score

Recent submission on 0:18 06 Mar 2020 Result: PASSED [PPPP]

Current Recent All

File Average Score

Read scores from the specified file. Then report the average value.

ให้เขียน โปรแกรมเพื่ออ่านข้อมูลคะแนนจากไฟล์ จากนั้นคำนวณหาค่าเฉลี่ยของคะแนนในไฟล์ดังกล่าว

โดยเขียนฟังก์ชัน `findAverage` ที่รับชื่อไฟล์ และคืนค่าเฉลี่ยของคะแนนในไฟล์

Example 1

Suppose the file `scores.txt` contains the following:

```
89
42
70
85
33
26
29
65
27
84
```

Then the program's output should be:

```
scores.txt
Average score is 55.0000
```

Example 2

The file `big-scores.txt` (click the link to download) contains 100 scores, which should produce the following output:

```
big-scores.txt
Average score is 64.6700
```

```
#include <stdio.h>

double findAverage(char *filename)
{

}

int main()
{
    char filename[20];
    double average;
    scanf("%s", filename);
    average = findAverage(filename);
    printf("Average score is %.4lf\n", average);
}
```

[hide line #]

- 01 - กำหนดโครงสร้างข้อมูล enum
- 02 - คำนวณวันในสัปดาห์
- 03 - Struct ที่เก็บพิกัดของจุดและหาระยะทาง
- 04 - Word Count
- 05 - Single Linked List
- 06 - จำนวนบิตที่ไม่ใช่ 0
- 07 - File Average Score
- 08 - Stock Log File

Current

Recent

All

Stock Log File

ตามูและตามิ สองพี่น้องเปิดร้านขายเสื้อผ้าขนาดใหญ่แถวประตูน้ำ ตามูเป็นคนใจดี เขียน โปรแกรมจัดการระบบขายเสื้อผ้าด้วยตัวเอง โปรแกรมของตามูจะเก็บบันทึกการขายเสื้อผ้าแยกไว้เป็นวันละไฟล์

รูปแบบของข้อมูลในไฟล์บันทึกการขายเสื้อผ้าของ โปรแกรมที่ตามูเขียนเป็นดังนี้

- บรรทัดที่ 1 เป็นจำนวนประเภทของเสื้อผ้าที่ขายได้ในวันนั้น
- บรรทัดที่ 2 เป็นต้นไป เป็นรายละเอียดของเวลาที่ขายเสื้อผ้า ประเภทเสื้อผ้าที่ขาย จำนวนที่ขาย และราคาที่ขาย มีรูปแบบคือ *ปีเดือนวัน,เวลา,ชื่อประเภทเสื้อผ้า,จำนวนที่ขาย,ราคาขาย*

ตามิเป็นคนขี้เหนียว เห็นที่ว่าตามูที่เป็นคนใจดี จะขายเสื้อผ้าได้กำไรน้อย เพราะตามูชอบใจอ่อนกับลูกค้าที่น่ารัก ๆ อยู่เสมอ จึงเขียน โปรแกรมเพื่อสรุปจำนวนและยอดขายของเสื้อผ้าแต่ละประเภทจากไฟล์บันทึกการขายเสื้อผ้าที่ได้จาก โปรแกรมที่ตามูเขียน

เมื่อวันที่ 1 เมษายน 2017 ที่ผ่านมา เป็นวันเมษาคนง ผู้คนเลยไม่ค่อยออกไปที่ไหน วันนั้นไฟล์บันทึกมีรายการน้อยมาก เหมาะแก่การนำมาเป็นไฟล์ตัวอย่างในการเขียน โปรแกรม ตามิจึงไปขอไฟล์ตามู และได้ไฟล์ซึ่งมีข้อมูลดังนี้

ข้อมูลในไฟล์ stock-log-20170401.txt

```
4
20170401,12:54:00,T Shirt L,1,120.00
20170401,12:56:32,Boxer,2,90.00
20170401,13:12:34,T Shirt L,2,200.00
20170401,13:12:34,Boxer,2,98.00
20170401,14:29:17,T Shirt M,4,480.00
20170401,16:39:27,Shorts M,1,120.50
20170401,17:12:34,T Shirt L,3,350.00
```

ตามิต้องการเขียน โปรแกรม เพื่อสรุปข้อมูลให้ได้ดังนี้

Product Name	Amount	Total
T Shirt L	6	670.00
Boxer	4	188.00
T Shirt M	4	480.00
Shorts M	1	120.50

จะเห็นว่า ผลลัพธ์เรียงลำดับประเภทสินค้าจากประเภทสินค้าแรกที่ปรากฏในไฟล์

จงช่วยตามิเขียน โปรแกรมเพื่อให้ได้ผลลัพธ์ตามความต้องการของตามิ โดยใน main จะรับชื่อไฟล์จาก standard input และส่งชื่อไฟล์นี้ไปให้ฟังก์ชัน reportStock

ตัวอย่างโปรแกรม

stock-log-20170401.txt

Product Name	Amount	Total
T Shirt L	6	670.00
Boxer	3	139.00
T Shirt M	4	480.00
Shorts M	1	120.50

[hide line #]

```
#include <stdio.h>
#include <string.h>

typedef struct _product {
    char name[30];
    int amount;
    double totalCost;
} Product;

void printStock(Product stock[], int nStock)
{
    int i;
    printf("%-20s|%6s|%-10s\n", "    Product Name", "Amount", "    Total");
    for (i = 0; i < nStock; i++) {
        printf("%-20s|%6d|%10.2lf\n",
            stock[i].name, stock[i].amount, stock[i].totalCost);
    }
}

int main()
{
    char filename[20];
    double average;
    scanf("%s", filename);
    reportStock(filename);
}
```