แบบฝึกหัดปฏิบัติการคาบที่ 12: Problem Solving

ชื่อ-นามสกุล.....หิรัญ สุขสมรัตน์.....รหัสประจำตัวนักศึกษา.....6404062610499.....

วันที่.....2...เดือน.....พฤษภาคม.....พ.ศ. 2565 Section.....3.....

1. [4G] จากการประมูลคลื่นระบบ 4G ที่ดุเดือด บริษัทให้บริการโทรศัพท์มือถือมีแนวโน้มที่จะออกแพ็คเกจบริการที่ซับซ้อน โดยมีการระบุว่าถ้าใช้แบบเติมเงินแล้วโทรตอนกลางวันจะคิดนาทีละ 0.75 บาท แต่ถ้าโทรตอนกลางคืนจะคิด 1.25 บาท ส่วนแบบจ่ายรายเดือนมีให้เลือก 2 ทางเลือกคือแบบ 300 บาทต่อเดือนและแบบ 600 บาทต่อเดือน โดยแบบ 300 บาทต่อเดือนจะโทรได้ 500 นาทีถ้าเกินนั้นจะคิดค่าโทรนาทีละ 1.50 บาท ส่วนแบบ 600 บาทต่อเดือนจะโทรได้ 1200 นาทีถ้าเกินนั้นจะคิดค่าโทรนาทีละ 1.25 บาท

นักศึกษาต้องการประหยัดค่าใช้จ่ายมากที่สุดจึงได้ทำการบันทึกว่าในแต่ละสัปดาห์ตนเองโทรตอนกลางวันกี่นาทีและตอนกลางคืนกี่นาที โดยจดบันทึกข้อมูลการใช้โทรศัพท์นี้เป็นเวลา 4 สัปดาห์

จงเขียนโปรแกรมรับค่าตัวเลขจำนวนการโทรตอนกลางวัน และตอนกลางคืน จากนั้นโปรแกรมจะพิมพ์เลข 1 ถ้าแบบเติมเงินมีค่าใช้จ่ายน้อยที่สุด พิมพ์เลข 2 ถ้าแบบ 300 บาทต่อเดือนมีค่าใช้จ่ายน้อยที่สุด และพิมพ์เลข - ถ้าแบบ 600 บาทต่อเดือนมีค่าใช้จ่ายน้อยที่สุด

ข้อมูลนำเข้า บรรทัดที่ 1 - 4 จำนวนการโทรตอนกลางวัน และตอนกลางคืน

ข้อมูลส่งออก พิมพ์เลขทางเลือกแพ็คเกจที่มีค่าใช้จ่ายน้อยที่สุด

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก	ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
100 100 100 100 100 100 100 100	3	50 20 60 70 40 30 50 50	2

```c
#include<stdio.h>

float top_up(float time[4][2])
{

    float timeday,timenight,total;

    timeday = time[0][0] + time[1][0] + time[2][0] + time[3][0] ;
    timenight = time[0][1] + time[1][1] + time[2][1] + time[3][1] ;

    total = (timeday*0.75) + (timenight*1.25) ;

return total ;
}

float monthly300(float time[4][2])
{

    int i,j;
    float sumtime = 0,total = 0 ;

    for(i=0;i<4;i++)
{
        for(j=0;j<2;j++)
    {
        sumtime = sumtime + time[i][j];
    }
}
    if(sumtime>500)
    {
        sumtime = sumtime - 500 ;
        total = sumtime*1.5 ;
    }
    else{}

return total + 300 ;
```

```
100 100
100 100
100 100
100 100
3
Process returned 0 (0x0)   execution time : 17.884 s
Press any key to continue.

50 20
60 70
40 30
50 50
2
Process returned 0 (0x0)   execution time : 8.743 s
Press any key to continue.
```

```
int main()
{

    int i,j;
    float time[4][2],x,y,z;

    for(i=0;i<4;i++)
{
        for(j=0;j<2;j++)
    {
        scanf("%f",&time[i][j]);
    }
}

x = top_up(time);
y = monthly300(time);
z = monthly600(time);
if(x < y && x < z){
    printf("1");
}
else if(y < x && y < z){
    printf("2");
}
else {
    printf("3");
}
return 0 ;
}
```

ชื่อ-นามสกุล.....หิรัญ สุขสมรัตน์.....รหัสประจำตัวนักศึกษา.....6404062610499.....

วันที่...2...เดือน.....พฤษภาคม.....พ.ศ. 2565 ตอนเรียน Lab ที่...3...

---

2. [Distance] กำหนดจุดในระนาบสามมิติมีตัวอย่างการเก็บในรูปแบบต่อไปนี้

float points[ ][ ] = {{-1, 0, 3}, {-1, -1, -1}, {4, 1, 1},{2, 0.5, 9}, {3.5, 2, -1},

{3, 1.5, 3}, {-1.5, 4, 2}, {5.5, 4, -0.5}};

จงเขียนโปรแกรมเพื่อคำนวนระยะทางระหว่างจุดสองสุดในระนาบสามมิติที่มีระยะทางระหว่างจุดมากที่สุด 3 อันดับแรก โดยระยะทางระหว่างสองจุด (x_1, y_1, z_1) และ (x_2, y_2, z_2) คำนวณได้จาก

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

ข้อมูลนำเข้า

บรรทัดแรก ระบุจำนวนตัวเลข n

บรรทัดที่ 2 ถึง n+1 ระบุจุดในระนาบสามมิติ

ข้อมูลส่งออก

บรรทัดที่ 1 - 3 แสดงระยะทางระหว่างจุดมากที่สุด 3 อันดับแรก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
8 -1 0 3 -1 -1 -1 4 1 1 2 0.5 9 3.5 2 -1 3 1.5 3 -1.5 4 2 5.5 4 -0.5	10.71 10.55 10.22

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

int main()
{
    int n,dis=0,i,j,k=0;
    scanf("%d",&n);
    for(i=n-1;i>=0;i--)
    {
        dis = dis + i;
    }
    float points[n][3],distance[dis],temp;

    for(i=0;i<n;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%f",&points[i][j]);
        }
    }


    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            distance[k] = sqrt(pow(points[i][0]-points[j][0],2)+pow(points[i][1]-points[j][1],2)+pow(points[i][2]-points[j][2],2));
            k++;
        }
    }

    for(i=1;i<dis;i++)
        {
        temp = distance[i];
        for(k=i-1; k >= 0 && distance[k]<temp; k--)
            {
            distance[k + 1] = distance[k];
            }
        distance[k+1] = temp;
        }

    printf("%.2f\n%.2f\n%.2f",distance[0],distance[1],distance[2]);
return 0;
}
```

```
8
-1 0 3
-1 -1 -1
4 1 1
2 0.5 9
3.5 2 -1
3 1.5 3
-1.5 4 2
5.5 4 -0.5
10.71
10.55
10.22
Process returned 0 (0x0)   execution time : 390.066 s
Press any key to continue.
```

3. [Visible Trees] มีต้นไม้ ความสูงต่างๆกัน เรียงเป็นแนวเส้นตรง เมื่อญาญ่า เดินผ่านต้นไม้แต่ละต้น ได้บันทึกความสูงของแต่ละต้นเอาไว้ ตามลำดับ จากนั้นเมื่อ**มองย้อนกลับไป** จะมีต้นไม้จำนวนหนึ่งเท่านั้น ที่สามารถมองเห็นได้ ในแนวเส้นตรงเดียวกัน เพราะต้นไม้ที่มีความสูงเท่ากันหรือต่ำกว่า จะถูกบดบัง จงหาว่า มีต้นไม้กี่ต้น ที่ญาญ่าจะสามารถมองเห็นได้

ข้อมูลนำเข้า

บรรทัดแรกคือค่า *n* (1<=*n*<=10) จำนวน test case

และในอีก *n* บรรทัดต่อมา แต่ละบรรทัดคือ หนึ่ง test case ซึ่งประกอบด้วย T (1<=*T*<=80) ระบุจำนวนต้นไม้และมีจำนวนเต็มบวกอีก *T* ค่า เป็นความสูงของต้นไม้แต่ละต้นที่บันทึกไว้ตามลำดับ

ข้อมูลส่งออก

แต่ละ Test case ให้แสดง จำนวนต้นไม้ที่สามารถมองเห็นได้

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
3 12 1 2 3 4 5 6 7 8 9 10 11 **12** 8 2 **13** 6 1 **7** 2 1 **3** 5 **15** 10 **10 9 8**	1 3 4

```c
#include<stdio.h>

int main()
{
    int n,m,i,j;
    scanf("%d",&n);
    int heightest=0;
    int count = 0;

    for(i=0;i<n;i++)
    {
        scanf("%d",&m);
        int array[m];
        for(j=0;j<m;j++)
        {
            scanf("%d",&array[j]);
        }
        for(j=m-1;j!=-1;j--){
            if(heightest < array[j])
            {
                heightest = array[j];
                count ++;
            }
        }
        printf("%d\n",count);
        count = 0;
        heightest = 0;
    }
}
```

```
3
12 1 2 3 4 5 6 7 8 9 10 11 12
1
8 2 13 6 1 7 2 1 3
3
5 15 10 10 9 8
4

Process returned 0 (0x0)   execution time :
Press any key to continue.
```

4. หน่วยสืบคดีพิเศษของประเทศแห่งหนึ่งต้องการค้นหาแหล่งกบดานของนักบวชรูปหนึ่ง โดยแหล่งที่พักของนักบวชรูปนี้มีความสลับซับซ้อนเป็นพิเศษ บุคคลภายนอกไม่สามารถเข้าถึงได้โดยตรง ด้วยเหตุนี้หน่วยสืบคดีพิเศษจึงจำเป็นต้องอาศัยอากาศยานไร้คนขับ หรือโดรน ทำการถ่ายภาพบริเวณที่สนใจ **โดยภายในภาพถ่ายจะปรากฏจำนวนคน ณ บริเวณที่กำหนด** โดยเป็นรูปขนาด HxW ช่อง ซึ่งหน่วยสืบคดีพิเศษต้องการหานักบวชจากรูปภาพนี้

ตัวอย่างของรูปขนาด 4X5 แสดงเป็นตารางด้านล่างกำหนดตารางชื่อ A ตัวเลขในแต่ละช่องแสดงจำนวนคนที่อยู่ในช่องนั้น

5	1	2	10	4
4	30	3	0	100
3	25	10	4	10
3	20	4	8	5

ในการหาตำแหน่งของนักบวชเนื่องจากเป็นนักบวชที่มีความสำคัญจึงจำเป็นต้องมีคนอยู่รอบข้าง ดังนั้นจึงมีมีเงื่อนไข 3 ข้อดังนี้

1. นักบวชจะปรากฏในบริเวณที่เป็น 2 ช่องติดกันพอดี
2. สองช่องที่เป็นบริเวณที่มีนักบวชควรมีจำนวนคน ณ บริเวณนั้นต่างกันไม่เกิน 10
3. เนื่องจากเป็นนักบวชที่มีความสำคัญจึงจำเป็นต้องมีคนอยู่รอบข้าง ตำแหน่งที่นักบวชอาศัยอยู่จึงน่าจะเป็นตำแหน่งที่มีจำนวนคน ณ บริเวณนั้นอยู่เป็นจำนวนมาก คือต้องเป็นสองช่องที่มีผลรวมของจำนวนคน ณ บริเวณนั้นอยู่เป็นจำนวนมาก

จากตารางตำแหน่งที่ตรงตามเงื่อนไขคือ A[2][2] และ A[3][2]

จงเขียนโปรแกรมที่รับตารางแสดงตำแหน่งของนักบวช จากนั้นให้หาตำแหน่งมุมบนซ้ายของช่องที่น่าจะปรากฏนักบวชมากที่สุด โดยระบุแถวและคอลัมภ์ช่องนั้น

ข้อมูลนำเข้า

บรรทัดแรก ระบุขนาดตาราง HxW

บรรทัดที่ 2 ถึง H+1 แสดงจำนวนคนในแถวที่ i โดยระบุเป็นจำนวนเต็มจำนวน W ตัว จำนวนที่ j จะเป็นจำนวนคนในช่องที่อยู่ในคอลัมภ์ j

ข้อมูลส่งออก

มีบรรทัดเดียว คือ **มุมบนซ้ายของช่องที่น่าจะปรากฏนักบวชมากที่สุด**โดยระบุแถวและคอลัมภ์ช่องนั้น

ตัวอย่างข้อมูลนำเข้า	**ตัวอย่างข้อมูลส่งออก**	**ตัวอย่างข้อมูลนำเข้า**	**ตัวอย่างข้อมูลส่งออก**
4 5 5 1 2 10 4 4 **30** 3 0 100 3 **25** 10 4 10 3 20 4 8 5	2 2	4 4 0 0 0 0 0 0 0 0 0 **1** 1 1 1 **1** 0 0	3 2

ชื่อ-นามสกุล.....หิรัญ สุขสมรัตน์.....รหัสประจำตัวนักศึกษา.....6404062610499.....

วันที่...2...เดือน.....พฤษภาคม.....พ.ศ. 2565 ตอนเรียน Lab ที่...3...

```c
#include<stdio.h>
int main()
{
    int i,j,num,num2,max=0,ans1,ans2;

    scanf("%d %d",&num,&num2);
    int A[num][num2];
    for (i=0;i<num;i++)
    {
        for(j=0;j<num2;j++)
        {
            scanf("%d",&A[i][j]);
        }
    }

    for (i=0;i<num;i++)
    {
        for(j=0;j<num2-2;j++)
        {
            if(A[i][j]-A[i][j+2]<=10||A[i][j]-A[i][j+2]<=-10)
            {
                if(max<A[i][j+1])
                {
                    max=A[i][j+1];
                    ans1=i+1;
                    ans2=j+2;
                }
            }
        }
    }
    printf("%d %d",ans1,ans2);
}
```

```
4 5
5 1 2 10 4
4 30 3 0 100
3 25 10 4 10
3 20 4 8 5
2 2
Process returned 0 (0x0)   execution time : 15.996 s
Press any key to continue.
```

```
4 4
0 0 0 0
0 0 0 0
0 1 1 1
1 1 0 0
3 2
Process returned 0 (0x0)   execution time : 15.780 s
Press any key to continue.
```

5. **[Line]** เส้นตรงคือการนำจุดสองจุดใดมาเชื่อมต่อกันโดยเส้นตรงจะประกอบด้วยสมาชิกที่เป็นจุดจำนวน 2 จุด คือจุดที่เป็นจุดเริ่มต้นของเส้นตรง (begin) และจุดที่เป็นจุดสุดท้ายของเส้นตรง (end) โดยมีโครงสร้างดังนี้

```
typedef struct{
  POINT begin;
  POINT end;
}LINE;

typedef struct{
   int x;
   int y;
}POINT;
```

จงเขียนโปรแกรมโดยการใช้ฟังก์ชันที่รับพารามิเตอร์ 2 ตัวที่มีชนิดข้อมูลเป็น POINT จากนั้นให้นำโครงสร้างดังกล่าวไปสร้างเป็นเส้นตรง (LINE) และคืนเป็นเส้นตรงออกมา หลังจากนั้นให้เขียนฟังก์ชันที่รับตัวแปรที่เป็น LINE เข้ามาในฟังก์ชันแล้วคืนเลข 1 2 หรือ 3 โดยที่

1 คือเส้นตรงที่มีลักษณะเป็นแนวตั้ง (Vertical)

2 คือเส้นตรงที่มีลักษณะเป็นแนวนอน (horizontal)

3 คือเส้นตรงที่ไม่ถูกนิยาม (oblique)

โดย Vertical line คือ เส้นตรงที่มีจุด begin กับจุด end มีพิกัด x อยู่ตำแหน่งเดียวกัน

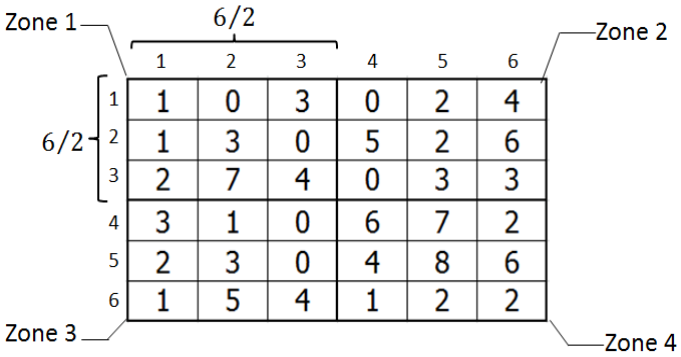
Horizontal line คือ เส้นตรงที่มีจุด begin กับจุด end มีพิกัด y อยู่ตำแหน่งเดียวกัน

Oblique line คือ เส้นตรงที่ไม่เป็นทั้ง vertical line หรือ horizontal line

```c
#include<stdio.h>

typedef struct{
    int x ;
    int y ;
}POINT ;

typedef struct{
    POINT begin ;
    POINT end ;
}LINE;

int what_line(LINE l){
    int ans ;
    if(l.begin.x == l.end.x){
        ans = 1;
    }
    else if(l.begin.y == l.end.y){
        ans = 2;
    }
    else ans = 3;
return ans;
}

int make_line(POINT a,POINT b){
    int ans ;
    LINE l;
    l.begin.x = a.x ;
    l.begin.y = a.y ;
    l.end.x = b.x ;
    l.end.y = b.y ;
    ans = what_line(l);
return ans ;
}
```

```c
int main(){
    int x ;
    POINT a,b;
    LINE l ;
    scanf("%d %d",&a.x,&a.y);
    scanf("%d %d",&b.x,&b.y);

    x = make_line(a,b);

    printf("%d",x);
}
```

6. [พื้นที่ในอาร์เรย์สองมิติ] อาร์เรย์ของเลขจำนวนเต็ม 2 มิติ ประกอบไปด้วย R แถว และ C คอลัมน์ โดยที่ R และ C เป็นเลขคู่จำนวนเต็มบวก. ถ้าต้องการแบ่งพื้นที่ในอาร์เรย์นี้ออกเป็น 4 ส่วน ได้แก่ zone 1, 2, 3 และ 4 โดยที่แต่ละโซนจะประกอบไปด้วยพื้นที่สี่เหลี่ยมจัตุรัสขนาด $R/2 \times C/2$ ช่องในอาร์เรย์ ตัวอย่างเช่น การแบ่งพื้นที่ของอาร์เรย์ขนาด 6 คูณ 6 แสดงได้ดังรูปด้านล่าง



	1	2	3	4	5	6
1	1	0	3	0	2	4
2	1	3	0	5	2	6
3	2	7	4	0	3	3
4	3	1	0	6	7	2
5	2	3	0	4	8	6
6	1	5	4	1	2	2

จงเขียนโปรแกรมเพื่อแสดงผลบวกที่มากที่สุดของสมาชิกในแต่ละโซน (maximum total sum) ตัวอย่างเช่น ผลบวกของโซน 1 ในอาร์เรย์ด้านบน คือ 1+0+3+1+3+0+2+7+4 = 21 ขณะที่ผลบวกของอาร์เรย์ในโซน 4 คือ 38

ข้อมูลเข้า

บรรทัดแรกเป็นจำนวนเต็มบวก R และ C

R บรรทัดต่อมาเป็นตัวเลขในอาร์เรย์แต่ละแถว โดยแต่ละแถวมี C คอลัมภ์

ข้อมูลส่งออก

ผลบวกที่มากที่สุดของสมาชิกในโซน

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลออก
2 4 1 2 3 4 5 6 7 8	15
4 2 1 2 3 4 5 6 7 8	14

ชื่อ-นามสกุล.....หิรัญ สุขสมรัตน์.....รหัสประจำตัวนักศึกษา.....6404062610499.....

วันที่...2...เดือน.....พฤษภาคม.....พ.ศ. 2565 ตอนเรียน Lab ที่...3...

```c
#include<stdio.h>

int main()
{
    int r,c,i,j;
    scanf("%d%d",&r,&c);

    int array[r][c];
    int zone1[r/2][c/2],zone2[r/2][c/2],zone3[r/2][c/2],zone4[r/2][c/2];
    int sum1=0,sum2=0,sum3=0,sum4=0;

    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&array[i][j]);
        }
    }

    for(i=0;i<r/2;i++)
    {
        for(j=0;j<c/2;j++)
        {
            sum1 = sum1 + array[i][j];
        }
    }

    for(i=0;i<r/2;i++)
    {
        for(j=c/2;j<c;j++)
        {
            sum2 = sum2 + array[i][j];
        }
    }

    for(i=r/2;i<r;i++)
    {
        for(j=0;j<c/2;j++)
        {
            sum3 = sum3 + array[i][j];
        }
    }

    for(i=r/2;i<r;i++)
    {
        for(j=c/2;j<c;j++)
        {
            sum4 = sum4 + array[i][j];
        }
    }
    if(sum1>sum2 && sum1>sum3 && sum1>sum4){printf("%d",sum1);}
    else if(sum2>sum1 && sum2>sum3 && sum2>sum4){printf("%d",sum2);}
    else if(sum3>sum1 && sum3>sum2 && sum3>sum4){printf("%d",sum3);}
    else {printf("%d",sum4);}

return 0 ;
}
```

```
2 4
1 2 3 4
5 6 7 8
15
Process returned 0 (0x0)   exe
Press any key to continue.
```

```
4 2
1 2
3 4
5 6
7 8
14
Process returned 0 (0x0)
Press any key to continue.
```