

## แบบฝึกหัดปฏิบัติการคาบที่ 6: Arrays

ชื่อ-นามสกุล.....รหัสประจำตัวนักศึกษา.....

วันที่.....เดือน.....พ.ศ. 2564

Section.....

1. จงเขียนผังงานและโปรแกรมเพื่อรับตัวเลขเข้ามา n จำนวน จากนั้นหาค่าเฉลี่ย หาค่ามากที่สุด ค่าน้อยสุด ค่า SD

### วิเคราะห์ปัญหา

### เขียนผังงาน

ข้อมูลนำเข้า

ข้อมูลส่งออก

กำหนดตัวแปร

ชื่อตัวแปร ชนิดตัวแปร ความหมาย

### เขียนโปรแกรม

```
# include <stdio.h>
# include <stdlib.h>
# include <math.h>
int main(){
    int n,i;
    float n_input;
    scanf("%i",&n);
    float total=0;
    for(i=0;i<n;i++){
        scanf("%f",&n_input);
        total += n_input;
    }
    printf("%.2f",total/n);
    return 0;
}
```

ชื่อ-นามสกุล.....รหัสประจำตัวนักศึกษา.....

วันที่.....เดือน.....พ.ศ. 2564

หมายเลขเครื่อง.....

2. จงเขียนโปรแกรมหาค่าฐานนิยม (Mode) ของตัวเลขจำนวนเต็ม 10 ตัว แล้วแสดงค่านี้ผ่านทางจอภาพ โดยกำหนดให้ตัวเลขแต่ละตัวต้องเป็นสมาชิกของเซต {5,10,15,20} เท่านั้น

ตัวอย่าง

Please enter 10 elements of integer in the set {5,10,15,20}

Element: 5 10 15 20 5 10 5 20 15 10

Mode = 5 10

### วิเคราะห์ปัญหา

ข้อมูลนำเข้า

ข้อมูลส่งออก

กำหนดตัวแปร

ชื่อตัวแปร    ชนิดตัวแปร    ความหมาย

### เขียนโปรแกรม

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(){
    int i,k,n1=0;
    int arrayed[] = {5,10,15,20},n[] = {0,0,0,0,0,0,0,0,0,0},total[] = {1,1,1,1,1,1,1,1,1,1},mod[] = {10,10};
    printf("Please enter 10 elements of integer in the set {5,10,15,20}\nElement: ");
    ; scanf("%i %i %i %i %i %i %i %i %i %i",&n[0],&n[1],&n[2],&n[3],&n[4],&n[5],&n[6],&n[7],&n[8],&n[9]);
    for (i=0;i<10;i++){
        for (k=0;k<4;k++){
            if (arrayed[k]==n[i]){
                for (l=i+1;l<10;l++){
                    if (n[l] == n[i] && i != 9){
                        total[i] += 1;
                    }
                    else{
                        continue;
                    }
                }
            }
            else {
                continue;
            }
        }
    }
    for (i=1;i<10;i++){
        if (n1 == 2){
            break;
        }
        else if (total[0]<total[i] || total[0]==total[i]){
            if (i==1 && total[0] == total[i]){
                mod[0]=0;
            }
            else{
                total[0] = total[i];
                mod[0] = i;
                n1 = 0;
            }
        }
        if (total[0] == total[i]){
            n1 += 1;
            mod[1] = i;
        }
    }

    if (total[mod[0]] == total[mod[1]] && n1 != 2){
        printf("Mode = %i %i",n[mod[0]],n[mod[1]]);
    } else if (total[mod[0]] != total[mod[1]]){
        printf("Mode = %i",n[mod[0]]);
    } else{
        printf("none mod");
    }
    return 0;
}
```

ชื่อ-นามสกุล.....รหัสประจำตัวนักศึกษา.....

วันที่.....เดือน.....พ.ศ. 2564

หมายเลขเครื่อง.....

3. ในการแข่งขันวิ่งระยะ 100 เมตร รายการหนึ่ง มีนักวิ่งแข่งขันทั้งสิ้น 10 คน จงเขียนโปรแกรมเพื่อรับอินพุตจากผู้ใช้งาน ได้แก่ ไอดีของนักวิ่ง (ID) และเวลาที่ใช้ในการวิ่ง (time) มีหน่วยเป็นวินาที จากนั้นโปรแกรมจะต้องแสดงสามอันดับแรกของนักวิ่ง (ID) ที่ใช้เวลาในการวิ่งน้อยที่สุดตามลำดับ กำหนดให้เวลามากที่สุดที่ใช้ในการวิ่ง 100 เมตร ต้องไม่เกิน 20.00 วินาที

## วิเคราะห์ปัญหา

ข้อมูลนำเข้า

ข้อมูลส่งออก

กำหนดตัวแปร

ชื่อตัวแปร    ชนิดตัวแปร    ความหมาย

## เขียนโปรแกรม

## เขียนผังงาน

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
struct runner {
    char id[100];
    int second;
};
int main(){
    int i,j,sc,r1=-1,r2=-1,r3=-1;
    char id;
    int index[] = {0,0,0,0,0,0,0,0,0,0};
    struct runner runners[10];
    for (i=0;i<10;i++){
        printf("Id%i: ",i);
        scanf("%s",runners[i].id);
        printf("timerunning[%i]<20.00: ",i);
        scanf("%i",&runners[i].second);
        printf("\n");
    }
    for (i=0;i<10;i++){
        index[i] = runners[i].second;
    }
    for(i=0;i<10;i++){
        for(j=1+i;j<10;j++){
            if(index[j] < index[i] && i!=9){
                sc = index[j];
                index[j] = index[i];
                index[i] = sc;
            }
            else{
                continue;
            }
        }
    }
    for (i=0;i<10;i++){
        if(runners[i].second == index[0]){
            index[0] = i;
        }else if(runners[i].second == index[1]){
            index[1] = i;
        }else if(runners[i].second == index[2]){
            index[2] = i;
        }else{
            continue;
        }
    }
    printf("1:%s 2:%s 3:%s",runners[index[0]].id,runners[index[1]].id,runners[index[2]].id);
    return 0;
}
```

#### 4. ตรวจสอบรหัสบัตรประชาชนอย่างไร

แบบฟอร์มในการสมัครสมาชิกของหลายๆ เว็บไซต์บังคับให้กรอกเลขที่บัตรประชาชน เช่น เว็บไซต์ของ, เว็บไซต์ประเภทเกมส์ออนไลน์ ฯลฯ ในการตรวจสอบเลขที่บัตรประชาชนนั้นทำได้โดยใช้ Check Digit หรือการตรวจสอบความถูกต้องโดยใช้ตัวเลขหลักสุดท้ายในการตรวจสอบ วิธีการ Check Digit มีดังนี้

1. ตัวเลขบนบัตรประชาชนจะมีทั้งหมด 13 หลัก นำเลขใน 12 หลักแรก มาคูณกับเลขประจำตำแหน่ง (เลขประจำหลักได้แก่ 13 บวก 1 ลบด้วยตำแหน่งที่)

จะได้ตัวเลขประจำตำแหน่งดังนี้

| ตำแหน่งที่ | เลขประจำตำแหน่ง |
|------------|-----------------|
| 1          | 13              |
| 2          | 12              |
| 3          | 11              |
| 4          | 10              |
| 5          | 9               |
| 6          | 8               |
| 7          | 7               |
| 8          | 6               |
| 9          | 5               |
| 10         | 4               |
| 11         | 3               |
| 12         | 2               |

2. หลังจากนั้นเอาผลคูณของทั้ง 12 หลักมารวมกัน แล้ว modulation (การหารเอาเศษ) ด้วย 11

3. เอาเศษที่ได้จากการหารในข้อ 2 มาลบด้วย 11 จะได้ Check Digit (ถ้าผลจากข้อ 2 ได้ 10 ให้เอาเลขหลักหน่วยเป็น Check Digit ก็คือ 0 นั่นเอง)

#### ตัวอย่าง

ต้องการเช็คว่ารหัสบัตรประชาชน 1234567890129 ถูกต้องหรือไม่ ทำได้โดย

1. นำตัวเลขคูณเลขประจำตำแหน่ง

$$(1 \times 13) + (2 \times 12) + (3 \times 11) + (4 \times 10) + (5 \times 9) + (6 \times 8) + (7 \times 7) + (8 \times 6) + (9 \times 5) + (0 \times 4) + (1 \times 3) + (2 \times 2) = 352$$

2. เอาผลคูณของทั้ง 12 หลักมารวมกัน แล้ว modulation (การหารเอาเศษ) ด้วย 11 จะได้

$$352 \% 11 = 0$$

3. นำ 11 ตั้งแล้วลบด้วย 0 จะได้

$$11 - 0 = 11 \text{ เอาเลขหลักหน่วย ดังนั้น Check Digit คือ } 1$$

4. นำ Check digit ที่ได้ในข้อ 3 ไปเปรียบเทียบกับรหัสบัตรประชาชนตำแหน่งที่ 13

เพราะฉะนั้นเลขที่บัตรประชาชน 1234567890129 ไม่ถูกต้อง ที่ถูกต้องคือ 1234567890121

จากขั้นตอนดังกล่าว จงเขียนโปรแกรมเพื่อรับค่าเลขรหัสประจำตัวประชาชนจากผู้ใช้งาน หลังจากนั้นให้ใช้ค่ารหัส

ดังกล่าวถูกต้องหรือไม่ โดยกรณีถูกต้องแสดงผลคำว่า Valid ส่วนถ้าไม่ถูกต้อง ให้แสดงผลคำว่า Invalid

เช่น

Please Enter ID: **3340100019856**

Your ID is **Valid**

### วิเคราะห์ปัญหา

### เขียนผังงาน

ข้อมูลนำเข้า

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

ข้อมูลส่งออก

```
int main(){
    char identify[14];
    int i,total=0;
    printf("Please Enter ID: ");
    scanf("%s",identify);
    for (i=0;i<12;i++){
        total= total + ((identify[i]%48)*(13-i));
    }
    total = total % 11;
    if(total==10){
        total = 0;
        if(0 == identify[12]%48){
            printf("Valid");
        }else{
            printf("Invalid");
        }
    } else if((identify[12]%48)==((11-total)%10)){
        printf("Valid");
    }else{
        printf("Invalid");
    }
    return 0;
}
```

กำหนดตัวแปร

ชื่อตัวแปร    ชนิดตัวแปร    ความหมาย

### เขียนโปรแกรม

การเขียนโปรแกรมส่งผ่าน Grader

1. [SD] จงเขียนโปรแกรมคำนวณส่วนเบี่ยงเบนมาตรฐาน ส่วนสูงของคน n คน รับข้อมูลส่วนสูงจากผู้ใช้นั้นคำนวณหาค่าส่วนเบี่ยงเบนมาตรฐาน

$$S.D. = \sqrt{\frac{(x - \bar{x})^2}{n-1}}$$

$x_i$  = ส่วนสูงคนที่  $i$

$\bar{x}$  = ส่วนสูงเฉลี่ยของข้อมูล

ข้อมูลอินพุต มี 2 บรรทัด บรรทัดแรกแสดงจำนวนคน n คน

บรรทัดที่ 2 รับข้อมูลส่วนสูงจากผู้ใช้นั้น n คน

ข้อมูลเอาต์พุต มี 1 บรรทัด แสดงค่าส่วนเบี่ยงเบนมาตรฐาน

ตัวอย่าง

| อินพุต                   | เอาต์พุต |
|--------------------------|----------|
| 5<br>160 178 169 177 168 | 7.37     |
| 5<br>185 166 172 167 177 | 7.83     |

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

int main(){
    int n=0,i,de;
    char ic[3];
    gets(ic);
    for (i=0;i<sizeof(ic);i++){
        if (ic[i+1]*1 != 0 ){
            n = n + (ic[i]%48)*(pow(10,1-i));
        }else{
            n = n+ ic[i]%48;
        }
    }
    float input[n],sd,xd=0,sd1=0;
    char input[100];
    gets(input);
    for (i=0;i<n;i++){
        input[i] = ((input[0+4*i]%48)*100) + ((input[1+(4*i)]%48)*10) + (input[2+(4*i)]%48);
    }
    printf("%d\n",n);
    for (i=0;i<n;i++){
        xd += input[i];
    }
    xd/=n;
    for(i=0;i<n;i++){
        sd1 = sd1 + pow((input[i]-xd),2);
    }
    sd = sqrt(sd1/(n-1));
    printf("%.2f",sd);

    return 0;
}
```

## 2. Occurrence of max number

จงเขียนโปรแกรมที่อ่านตัวเลขจำนวนจริง หาตัวเลขที่มีขนาดใหญ่มากที่สุดจากกลุ่มของตัวเลขดังกล่าว นับจำนวนครั้งของการปรากฏเลขดังกล่าว โดยให้ตัวเลข Input จบด้วยเลข 0 ตัวอย่างเช่น ถ้าป้อนตัวเลขต่อไปนี้ 3 5 2 5 5 0 โปรแกรมจะค้นหาจำนวนตัวเลขที่มากที่สุดคือ 5 และจำนวนครั้งของการเกิดเลข 5 คือ 4 เช่น

ข้อมูลนำเข้า รับข้อมูลเลขจำนวนเต็มบวก n ตัว

ข้อมูลส่งออก ตัวเลขที่มีค่ามากที่สุดและจำนวนครั้งของการปรากฏ

| ข้อมูลนำเข้า | ข้อมูลส่งออก |
|--------------|--------------|
| 3 5 2 5 5 0  | 5 4          |

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

int main(){
    int n=0,i=-1,total=1;
    while (i != 0){
        scanf("%i",&i);
        if(i==0){
            break;
        }
        if(i == n){
            total += 1;
        }
        if (i > n){
            n=i;
            total = 1;
        }
    }
    printf("%i %i",n,total);
    return 0;
}
```

3. [Matrix Transpose] จงเขียนโปรแกรมเพื่อคำนวณหาค่าทรานโพสของเมตริกซ์ (Matrix Transpose) A ขนาด  $n \times n$  คือ  $A^T_{n \times n}$

A =

|   |    |   |   |
|---|----|---|---|
| 1 | 5  | 3 | 7 |
| 2 | 6  | 9 | 2 |
| 4 | 10 | 1 | 0 |
| 9 | 0  | 8 | 7 |

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

$A^T =$

|   |   |    |   |
|---|---|----|---|
| 1 | 2 | 4  | 9 |
| 5 | 6 | 10 | 0 |
| 3 | 9 | 1  | 8 |
| 7 | 2 | 0  | 7 |

```
int main(){
    int n,i,j,k;
    scanf("%i %i",&n,&j);
    int matrix[i][j];
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&matrix[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("%d ",matrix[j][i]);
        }
        printf("\n");
    }
    return 0;
}
```

ข้อมูลอินพุต มี n+1 บรรทัด บรรทัดแรกเป็นขนาดของเมตริกซ์

บรรทัดที่ 2-+1 ประกอบสมาชิกของเมตริกซ์ A ขนาด  $n \times n$

ข้อมูลเอาต์พุต มี n บรรทัด ประกอบสมาชิกของเมตริกซ์  $A^T$  ขนาด  $n \times n$

ตัวอย่าง

| อินพุต   | เอาต์พุต |
|----------|----------|
| 4 4      | 1 2 4 9  |
| 1 5 3 7  | 5 6 10 0 |
| 2 6 9 2  | 3 9 1 8  |
| 4 10 1 0 | 7 2 0 7  |
| 9 0 8 7  |          |

4. [Matrix Addition] จงเขียนโปรแกรมเพื่อคำนวณหาผลบวกของเมตริกซ์ (Matrix Addition) ขนาด  $n \times n$  คือ

$$C_{n \times n} = A_{n \times n} + B_{n \times n} \quad \text{เช่น } n=4$$

A =

|   |    |   |   |
|---|----|---|---|
| 1 | 5  | 3 | 7 |
| 2 | 6  | 9 | 2 |
| 4 | 10 | 1 | 0 |
| 9 | 0  | 8 | 7 |

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

B =

|    |    |    |   |
|----|----|----|---|
| 11 | 3  | 2  | 1 |
| 5  | -5 | 0  | 4 |
| 12 | 1  | 1  | 3 |
| 7  | 5  | -4 | 6 |

ข้อมูลอินพุต มี n+1 บรรทัด บรรทัดแรกแสดงจำนวนของแถวและคอลัมน์ของเมตริกซ์ทั้งสอง

บรรทัดที่ 2 ถึง n+2 รับข้อมูลของเมตริกซ์ A และ B

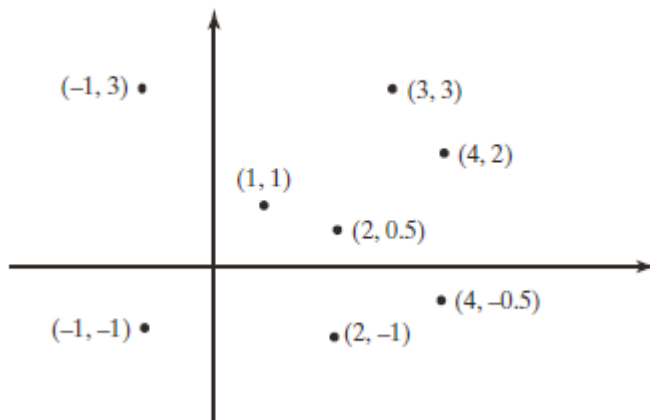
ข้อมูลเอาต์พุต มี n บรรทัด แสดงผลบวกของเมตริกซ์ (Matrix Addition) ขนาด  $n \times n$

ตัวอย่าง

| อินพุต   | เอาต์พุต  |
|----------|-----------|
| 4 4      | 12 8 5 8  |
| 1 5 3 7  | 7 1 9 6   |
| 2 6 9 2  | 16 1 2 13 |
| 4 10 1 0 | 16 5 4 13 |
| 9 0 8 7  |           |
| 11 3 2 1 |           |
| 5 -5 0 4 |           |
| 12 1 1 3 |           |
| 7 5 -4 6 |           |

```
int main(){
    int n,i,j,k;
    scanf("%i %i",&n,&k);
    int matrix1[i][j];
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&matrix1[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&matrix2[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            summatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            printf("%d ",summatrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

5. [Nearest point] จงเขียนโปรแกรมเพื่อคำนวณหาจุดที่ใกล้เคียงกันมากที่สุด



ข้อมูลอินพุต มี  $n+1$  บรรทัด บรรทัดแรกแสดงจำนวนของจุด

บรรทัดที่  $2 - n+1$  รับข้อมูลของจุดแต่ละจุด

ข้อมูลเอาต์พุต มี  $n$  บรรทัด แสดงคู่จุดที่ใกล้เคียงกันมากที่สุด และระยะห่างระหว่างจุดดังกล่าว  
ตัวอย่าง

| อินพุต | เอาต์พุต |
|--------|----------|
| 8      | 3 5 1.12 |
| -1 3   |          |
| 3 3    |          |
| 1 1    |          |
| 4 2    |          |
| 2 0.5  |          |
| 4 -0.5 |          |
| 2 -1   |          |
| -1 -1  |          |

```
# include <stdio.h># include <stdlib.h># include <math.h>int main(){ int n,i,k; scanf("%i",&
```



6. [MH320] ในวันปีใหม่ สนามกีฬาแห่งหนึ่งได้ประดับไฟที่พื้นสนามฟุตบอลเพื่อความสวยงาม ในการประดับไฟนั้นทำโดยแบ่งสนามสี่เหลี่ยมเป็นช่องย่อยๆ จำนวน  $N$  แถว แถวละ  $M$  คอลัมน์ รวม  $N \times M$  ช่อง เจ้าของสนามได้เปิดสนามให้ประชาชนทั่วไปเข้าชมเพื่อความสวยงาม

อย่างไรก็ตาม โลกนี้ไม่มีอะไรฟรี เจ้าของสนามจะต้องจ่ายค่าไฟให้กับไฟประดับเหล่านี้ เนื่องจากมีการประดับไฟเป็นลวดลายต่างๆ ค่าไฟของไฟแต่ละช่องไม่จำเป็นต้องเท่ากัน

เพื่อไม่ให้เป็นภาระขาดทุน เจ้าของสนามจึงได้จัดเครื่องโพนเจ็ตส่วนบุคคลให้กับประชาชนเช่าเพื่อเปิดไฟประดับ เครื่องโพนเจ็ตแต่ละเครื่องเมื่อเช่าไปแล้วจะผู้ใช้จะสามารถเปิดได้ทั้งสิ้น  $K$  ครั้ง ในการเปิดแต่ละครั้งจะใช้เชื้อเพลิงมูลค่าเท่ากับ  $L$  บาท ดังนั้น ค่าใช้จ่ายรวมทั้งหมดของเจ้าของสนามคือค่าไฟรวมของไฟประดับและค่าเชื้อเพลิงรวมของการเปิดเครื่องโพนเจ็ตในการเปิดทั้งหมด

เจ้าของสนามทราบว่าจะมีคนมาชมและเช่าเครื่องโพนเจ็ตจำนวน  $C$  คน เขาต้องการคำนวณค่าเช่าเครื่องโพนเจ็ตต่อคนที่น้อยที่สุด ที่จะทำให้เขาไม่ขาดทุน เพื่อให้การเช่าเป็นไปได้สะดวก ค่าเช่าจะต้องเป็นจำนวนเต็มเสมอด้วย

เขียนโปรแกรมรับราคาค่าไฟ ของสนามแต่ละช่อง รวมทั้งข้อมูลของการใช้เครื่องโพนเจ็ต จากนั้นคำนวณหาค่าเช่าเครื่องโพนเจ็ตต่อคนที่เป็จำนวนเต็มที่น้อยที่สุด ที่จะทำให้เจ้าของสนามไม่ขาดทุน

### ข้อมูลอินพุต

บรรทัดแรกระบุจำนวนเต็มบวก  $N$  และ  $M$  คั่นด้วยช่องว่าง แทนขนาดความกว้างและความยาวของสนาม ( $1 \leq N \leq 100$ ;  $1 \leq M \leq 100$ )

บรรทัดที่สองระบุจำนวนเต็มบวก  $L$  และ  $K$  คั่นด้วยช่องว่าง โดยที่  $L$  แทนราคาเชื้อเพลิงต่อการเปิดหนึ่งครั้งและ  $K$  แทนจำนวนครั้งที่เครื่องโพนเจ็ตใช้เปิดได้ต่อคนเช่าหนึ่งคน ( $1 \leq L \leq 100$ ;  $1 \leq K \leq 100$ )

บรรทัดที่สามระบุจำนวนเต็มบวก  $C$  แทนจำนวนผู้เล่นทั้งหมดที่เข้ามาเล่น ( $1 \leq C < 1,000$ )

บรรทัดถัดไปอีก  $N$  บรรทัด แต่ละบรรทัดรับจำนวนเต็มบวก  $M$  ตัว แต่ละตัวถูกคั่นด้วยช่องว่าง แทนค่าไฟในแต่ละช่องที่ประดับไฟ ซึ่งจะเป็นจำนวนเต็มบวกที่มีค่าไม่เกิน 3,000

### ข้อมูลเอาต์พุต

มีบรรทัดเดียวเป็นจำนวนเต็มบวกหนึ่งจำนวน แทนค่าเช่าเครื่องโพนเจ็ตต่อคนที่เป็จำนวนเต็มที่น้อยที่สุด ที่จะทำให้เจ้าของสนามไม่ขาดทุน

ชื่อ-นามสกุล.....รหัสประจำตัวนักศึกษา.....

วันที่.....เดือน.....พ.ศ. 2564

หมายเลขเครื่อง.....

### ตัวอย่าง

| อินพุต   | เอาต์พุต |
|--|----------|
| 3 3<br>2 1<br>1<br>1 1 1<br>1 1 1<br>1 1 1       | 11       |
| 3 4<br>3 2<br>7<br>1 2 3 4<br>4 3 2 1<br>1 1 1 1 | 10       |

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

int main(){
    int N,M,L,K,C,total=0,i,k,ans;
    scanf("%d %d",&N,&M);
    int calculate[N][M];
    scanf("%d %d",&L,&K);
    scanf("%d",&C);
    for (i=0;i<N;i++){
        for(k=0;k<M;k++){
            scanf("%i",&calculate[N][M]);
            total += calculate[N][M];
        }
    }
    total = total + (L*K*C);
    if (total % C != 0){
        ans = total/C;
        ans += 1;
    }else{
        ans = total/C;
    }
    printf("%i",ans);
    return 0;
}
```