

# แบบฝึกหัดปฏิบัติการคำบทที่ 6: Arrays

ชื่อ-นามสกุล ทํองทํัน

จำลอง

รหัสประจำตัวนักศึกษา ๖๖๐๔๐๖๒๖๓๖๒๐๘

วันที่ ๑๐ เดือน มกราคม พ.ศ. ๒๕๖๕

Lab. ๖ Section. ๔

- จงเขียนโปรแกรมเพื่อรับตัวเลขเข้ามา  $n$  จำนวน จากนั้นหาค่าเฉลี่ย หากค่ามากสุด ค่าน้อยสุด ค่า SD

## วิเคราะห์ปัญหา

ข้อมูลนำเข้า

$n$  จำนวนสมาชิก

$A[n]$  ค่าที่ศูนย์ซึ่งมีความสนใจ

ข้อมูลส่องออก

average

Min, Max

SD

กำหนดตัวแปร

ชื่อตัวแปร ชนิดตัวแปร

$n$  int

ความหมาย

จำนวนสมาชิก

average float

ค่าเฉลี่ย

SD float

จำนวนหน่วยเมตรฐาน

sum float

ผลรวม

sum1 float

ผลรวมที่สอง

variance float

ความแปรปรวน

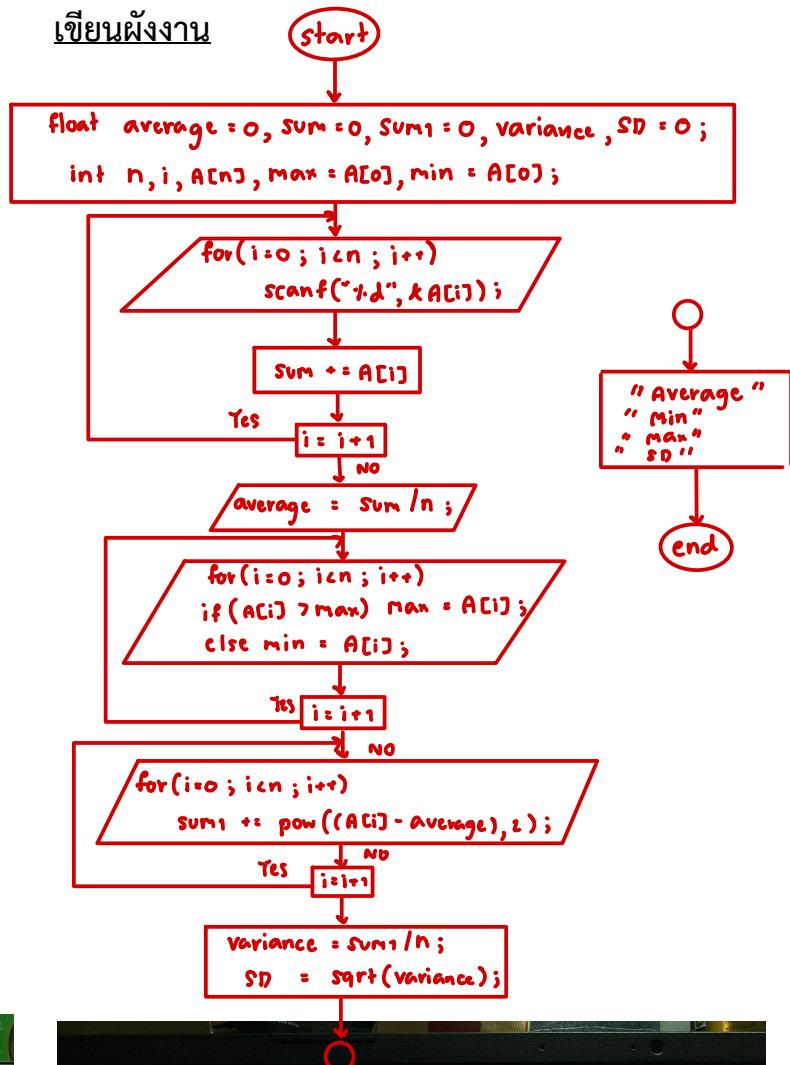
max int

ค่ามากที่สุด

min int

ค่าน้อยที่สุด

## เขียนโปรแกรม



Screenshot of the Dev-C++ IDE showing the code and its output.

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int main(){
5     int n;
6     float average = 0, SD = 0, sum = 0, sum1 = 0, variance;
7     scanf("%d", &n);
8     int A[n];
9     for(int i=0; i<n; i++){
10         scanf("%d", &A[i]);
11         sum = sum + A[i];
12     }
13     average = sum / n;
14     int max = A[0], min = A[0];
15     for(int i=0; i<n; i++){
16         if(A[i] > max){
17             max = A[i];
18         } else {
19             min = A[i];
20         }
21     }
22 }
  
```

Output window:

```

10
1 5 6 8 2 1 34 100 90 200
Average : 44.70
Maximum : 200
Minimum : 90
Standard Deviation : 62.77
  
```

Screenshot of the Dev-C++ IDE showing the completed code.

```

18     } else {
19         min = A[i];
20     }
21 }
22 for(int i=0; i<n; i++){
23     sum1 = sum1 + pow((A[i] - average), 2);
24 }
25 variance = sum1 / n;
26 SD = sqrt(variance);
27 printf("Average : %.2f\n", average);
28 printf("Maximum : %d\n", max);
29 printf("Minimum : %d\n", min);
30 printf("Standard Deviation : %.2f\n", SD);
31 }
32
33
34
35
36
37
  
```

2. ในการแข่งขันวิ่งระยะ 100 เมตร รายการหนึ่ง มีนักวิ่งแข่งขันทั้งสิ้น 10 คน จะเขียนโปรแกรมเพื่อรับอินพุตจากผู้ใช้งานได้แก่ ไอดีของนักวิ่ง (ID) และเวลาที่ใช้ในการวิ่ง (time) มีหน่วยเป็นวินาที จากนั้นโปรแกรมจะต้องแสดงสามอันดับแรกของนักวิ่ง (ID) ที่ใช้เวลาในการวิ่งน้อยที่สุดตามลำดับ กำหนดให้เวลาหากว่าที่สุดที่ใช้ในการวิ่ง 100 เมตร ต้องไม่เกิน 20.00 วินาที

### วิเคราะห์ปัญหา

ข้อมูลนำเข้า

id time

ข้อมูลส่งออก

Top 3 id time

กำหนดตัวแปร

ชื่อตัวแปร

ชนิดตัวแปร

ความหมาย

id

int

ตัวบัญชี

time

double

เวลาที่ใช้

MAX\_RUNNERS

int

จำนวนคนวิ่งทั้งหมด

### เขียนโปรแกรม

```

1 #include <stdio.h>
2
3 #define MAX_RUNNERS 10
4
5 int main() {
6     int id[MAX_RUNNERS];
7     double time[MAX_RUNNERS];
8
9     printf("Enter runner data (ID and time for each runner):\n");
10    for(int i = 0; i < MAX_RUNNERS; i++) {
11        printf("Runner %d - Enter ID and time: ", i+1);
12        scanf("%d %f", &id[i], &time[i]);
13
14        if(time[i] > 20.00) {
15            printf("Invalid time. Must be <= 20.00 seconds\n");
16            i--;
17            continue;
18        }
19
20    }
21
22    for(int i = 0; i < MAX_RUNNERS-1; i++) {
23        if(time[i] > time[i+1]) {
24            double tempTime = time[i];
25            time[i] = time[i+1];
26            time[i+1] = tempTime;
27
28            int tempId = id[i];
29            id[i] = id[i+1];
30            id[i+1] = tempId;
31
32        }
33
34    }
35
36    printf("\nTop 3 runners:\n");
37    for(int i = 0; i < 3; i++) {
38        printf("%d: Runner ID %d - %.2f seconds\n", i+1, id[i], time[i]);
39    }
40
41
42

```

```

22
23    for(int j = 0; j < MAX_RUNNERS-i-1; j++) {
24        if(time[j] > time[j+1]) {
25            double tempTime = time[j];
26            time[j] = time[j+1];
27            time[j+1] = tempTime;
28
29            int tempId = id[j];
30            id[j] = id[j+1];
31            id[j+1] = tempId;
32
33        }
34
35    }
36    printf("\nTop 3 runners:\n");
37    for(int i = 0; i < 3; i++) {
38        printf("%d: Runner ID %d - %.2f seconds\n", i+1, id[i], time[i]);
39    }
40
41
42

```

### 3. ตรวจสอบรหัสบัตรประชาชนอย่างไร

แบบฟอร์มในการสมัครสมาชิกของหลายๆ เว็บไซต์บังคับให้กรอกเลขที่บัตรประชาชน เช่น เว็บขายของ, เว็บประเภทเกมส์ออนไลน์ฯลฯ ในการตรวจสอบเลขที่บัตรประชาชนนั้นทำได้โดยการใช้ 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 ให้อาเลขอหลักหน่วยเป็น Check Digit ก็คือ 0 นั้นเอง)

ตัวอย่าง

ต้องการเช็คว่ารหัสบัตรประชาชน 1234567890129 ถูกต้องหรือไม่ ทำได้โดย

1.นำตัวเลขคูณเลขประจำตัวหนึ่ง

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

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 ไปเปรียบเทียบกับรหัสบัตรประชาชนตัวหนึ่งที่ 13

เพราะฉะนั้นเลขที่บัตรประชาชน 1234567890129 ไม่ถูกต้อง ที่ถูกต้องคือ 1234567890121

จากขั้นตอนดังกล่าว จงเขียนโปรแกรมเพื่อรับค่าเลขรหัสประจำตัวประชาชนจากผู้ใช้ หลังจากนั้นให้เช็คว่ารหัส ดังกล่าวถูกต้องหรือไม่ โดยกรณีถูกต้องแสดงผลคำว่า Valid ส่วนถ้าไม่ถูกต้อง ให้แสดงผลคำว่า Invalid เช่น

Please Enter ID: **3340100019856**

Your ID is **Valid**

### วิเคราะห์ปัญหา

ข้อมูลนำเข้า

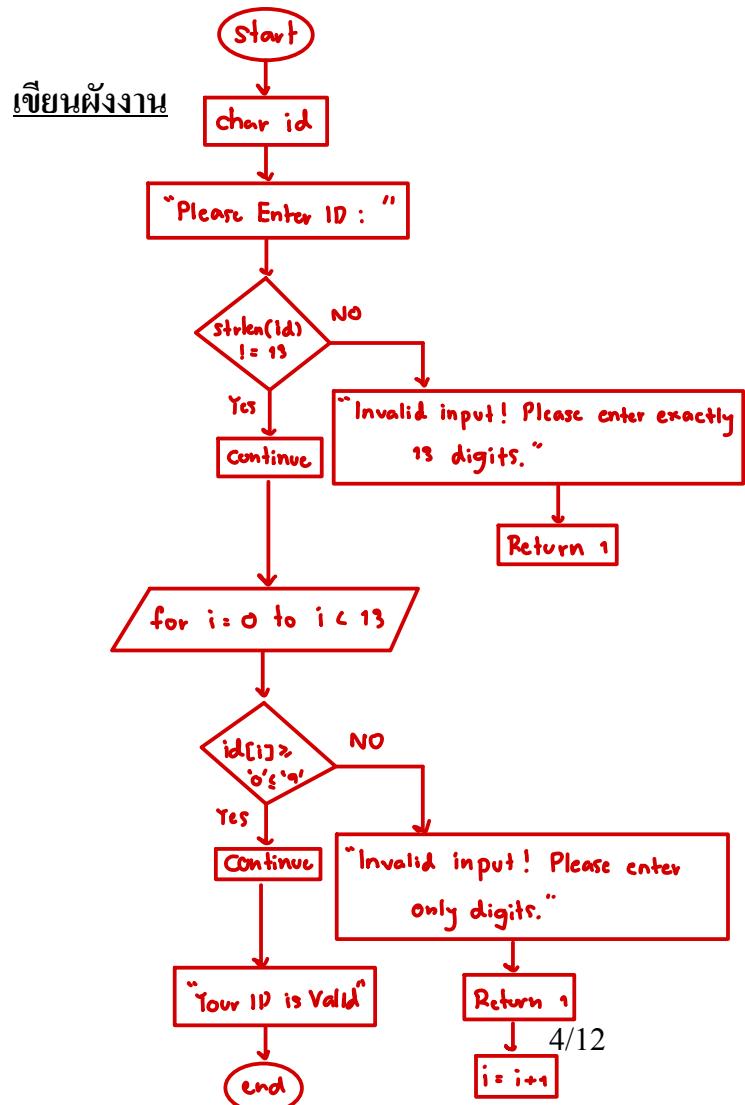
**Please Enter Id :**

ข้อมูลส่งออก

**Your Id is valid/invalid.**

กำหนดตัวแปร

ชื่อตัวแปร	ชนิดตัวแปร	ความหมาย
<b>id</b>	<b>char</b>	เลขบัตรประชาชน
<b>weights</b>	<b>int</b>	เลขประจำตัวหนึ่ง
<b>sum</b>	<b>int</b>	คำนวณผล



เขียนโปรแกรม

```

Structure\b3.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
I1.c I2.c I3.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char id[14];
6     int weights[12] = {13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2};
7     int sum = 0;
8
9     printf("Please Enter ID: ");
10    scanf("%s", id);
11
12    if (strlen(id) != 13) {
13        printf("Invalid input! Please enter exactly 13 digits.\n");
14        return 1;
15    }
16
17    for (int i = 0; i < 13; i++) {
18        if (id[i] < '0' || id[i] > '9') {
19            printf("Invalid input! Please enter only digits\n");
20            return 1;
21        }
22
23        for (int i = 0; i < 12; i++) {
24            sum += (id[i] - '0') * weights[i];
25        }
26
27        int check_digit = (11 - (sum % 11)) % 10;
28
29        if (check_digit == (id[12] - '0')) {
30            printf("Your ID is Valid\n");
31        } else {
32            printf("Your ID is Invalid\n");
33        }
34
35        return 0;
36    }
37

```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\שולחןทำงาน\C++\Structure\b3.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.33s

Set 0 Lines: 42 Length: 845 Insert Done parsing in 0.016 seconds

```

Structure\b3.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
I1.c I2.c I3.c
17    for (int i = 0; i < 13; i++) {
18        if (id[i] < '0' || id[i] > '9') {
19            printf("Invalid input! Please enter only digits\n");
20            return 1;
21        }
22
23        for (int i = 0; i < 12; i++) {
24            sum += (id[i] - '0') * weights[i];
25        }
26
27        int check_digit = (11 - (sum % 11)) % 10;
28
29        if (check_digit == (id[12] - '0')) {
30            printf("Your ID is Valid\n");
31        } else {
32            printf("Your ID is Invalid\n");
33        }
34
35        return 0;
36    }
37

```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\שולחןทำงาน\C++\Structure\b3.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.33s

Abort Compilation Shorten compiler paths

การเขียนโปรแกรมส่งผ่าน Grader

1. [SD] จะเขียนโปรแกรมคำนวณส่วนเบี่ยงเบนมาตรฐาน ส่วนสูงของคน  $n$  คน รับข้อมูลส่วนสูงจากผู้ใช้ จากนั้นคำนวณหาค่าส่วนเบี่ยงเบนมาตรฐาน

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

$x_i$  = ส่วนสูงคนที่  $i$

$\bar{x}$  = ส่วนสูงเฉลี่ยของข้อมูล

ข้อมูลอินพุท มี 2 บรรทัด บรรทัดแรกแสดงจำนวนคน  $n$  คน

บรรทัดที่ 2 รับข้อมูลส่วนสูงจากผู้ใช้  $n$  คน

ข้อมูลเอาท์พุท มี 1 บรรทัด แสดงค่าส่วนเบี่ยงเบนมาตรฐาน

ตัวอย่าง

อินพุท	เอาท์พุท
5 160 178 169 177 168	7.37
5 185 166 172 167 177	7.83

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int n;
6     scanf("%d", &n);
7     float heights[n];
8     float sum = 0.0, mean;
9     for(int i = 0; i < n; i++) {
10         scanf("%f", &heights[i]);
11         sum += heights[i];
12     }
13     mean = sum / n;
14     float sumSquareDiff = 0;
15     for(int i = 0; i < n; i++) {
16         float diff = heights[i] - mean;
17         float squareDiff = diff * diff;
18         sumSquareDiff += squareDiff;
19     }
20     float variance = sumSquareDiff / (n-1);
21     float sd = sqrt(variance);
22     printf("%.2f", sd);
23     return 0;
24 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output File Name: C:\Users\user\appData\Local\Temp\Dev-C++\Structure\SD.exe
- Output Size: 132.8701171875 Kib
- Compilation Time: 0.19s

```

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output File Name: C:\Users\user\appData\Local\Temp\Dev-C++\Structure\SD.exe
- Output Size: 132.8701171875 Kib
- Compilation Time: 0.19s

ชื่อ-นามสกุล..... **ກວດກົມ** ລ້າກອນ

รหัสประจำตัวนักศึกษา..... **6604062636208**

วันที่..... **10** เดือน..... **ມັງກອນ** พ.ศ. 2562

Lab..... **6** หมายเลขอเร่ย়..... **—**

## 2. [Matrix Transpose] จงเขียนโปรแกรมเพื่อคำนวณหาค่า transpose ของเมตริกซ์ (Matrix Transpose)

A ขนาด  $n \times n$  คือ  $A^T$   $n \times n$

$$A = \begin{matrix} 1 & 5 & 3 & 7 \\ 2 & 6 & 9 & 2 \\ 4 & 10 & 1 & 0 \\ 9 & 0 & 8 & 7 \end{matrix}$$

$$A^T = \begin{matrix} 1 & 2 & 4 & 9 \\ 5 & 6 & 10 & 0 \\ 3 & 9 & 1 & 8 \\ 7 & 2 & 0 & 7 \end{matrix}$$

ข้อมูลอินพุท มี  $n+1$  บรรทัด บรรทัดแรกเป็นขนาดของเมตริกซ์

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

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

ตัวอย่าง

อินพุท	เอาท์พุท
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	

```

#include <stdio.h>
int main() {
    int a[10][10], transpose[10][10], r, c;
    scanf("%d %d", &r, &c);
    for(int i = 0; i < r; ++i){
        for(int j = 0; j < c; ++j) {
            if (j == c - 1);
            scanf("%d", &a[i][j]);
        }
    }
    for(int i = 0; i < r; ++i){
        for(int j = 0; j < c; ++j) {
            if (j == c - 1);
            for (int k = 0; k < r; ++k) {
                transpose[j][i] = a[k][j];
            }
        }
    }
    printf("\n");
}

```

Process exited after 8.058 seconds.  
Press any key to continue . . .

```

for(int i = 0; i < r; ++i){
    for(int j = 0; j < c; ++j){
        if (j == c - 1);
    }
}
for (int i = 0; i < r; ++i){
    for (int j = 0; j < c; ++j) {
        transpose[j][i] = a[i][j];
    }
}
printf("\n");
for (int i = 0; i < c; ++i){
    for (int j = 0; j < r; ++j) {
        printf("%d ", transpose[i][j]);
        if (j == r - 1)
            printf("\n");
    }
}
return 0;
}

```

Compilation results...

- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\user\www\file\C++\Structure\matrix\_trans.exe  
- Output Size: 129.2734375 Kib

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

$$C_{nxn} = A_{nxn} + B_{nxn}$$

เช่น  $n=4$

$$\begin{array}{r} A = \\ \begin{matrix} 1 & 5 & 3 & 7 \\ 2 & 6 & 9 & 2 \\ 4 & 10 & 1 & 0 \\ 9 & 0 & 8 & 7 \end{matrix} \end{array}$$

$$\begin{array}{r} B = \\ \begin{matrix} 11 & 3 & 2 & 1 \\ 5 & -5 & 0 & 4 \\ 12 & 1 & 1 & 3 \\ 7 & 5 & -4 & 6 \end{matrix} \end{array}$$

ข้อมูลอินพุท มี  $n+1$  บรรทัด บรรทัดแรกแสดงจำนวนของแถวและคอลัมน์ของเมตริกซ์ทั้งสอง

บรรทัดที่ 2 ถึง  $n*2$  รับข้อมูลของเมตริกซ์ A และ B

ข้อมูลเอาท์พุท มี  $n$  บรรทัด แสดงผลบวกของเมตริกซ์(Matrix Addition) ขนาด  $nxn$

### ตัวอย่าง

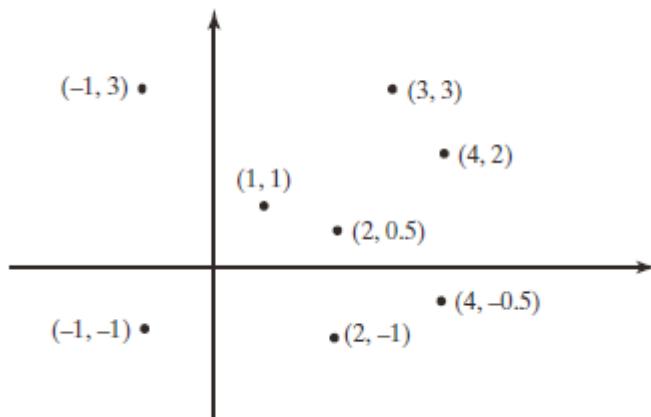
อินพุท	เอาท์พุท
4 4	12 8 5 8
1 5 3 7	7 1 9 6
2 6 9 2	16 11 2 3
4 10 1 0	16 5 4 13
9 0 8 7	
11 3 2 1	
5 -5 0 4	
12 1 1 3	
7 5 -4 6	

```

#include <stdio.h>
int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;
    scanf("%d %d", &r, &c);
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
            scanf("%d", &a[i][j]);
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
            scanf("%d", &b[i][j]);
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
            sum[i][j] = a[i][j] + b[i][j];
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j)
            printf("%d ", sum[i][j]);
    if (j == c - 1)
        printf("\n");
}
return 0;

```

4. [Nearest point] จงเขียนโปรแกรมเพื่อคำนวณหาค่าจุดที่ใกล้เคียงกันมากที่สุด



ข้อมูลอินพุท มี  $n+1$  บรรทัด บรรทัดแรกแสดงจำนวนของจุด

บรรทัดที่  $2 - n+1$  รับข้อมูลของจุดแต่ละจุด

ข้อมูลเอาท์พุท มี  $n$  บรรทัด แสดงคู่จุดที่ใกล้เคียงกันมากที่สุด และระยะห่างระหว่างจุดดังกล่าว

ตัวอย่าง

อินพุท	เอาท์พุท
8	3 5 1.12
-1 3	
3 3	
1 1	
4 2	
2 0.5	
4 -0.5	
2 -1	
-1 -1	

```
g++ C:\Users\user\OneDrive\桌面\C++\Structure\nearest_point.c -o nearest_point
Project Execute Tools AStyle Window Help
File Edit View Insert Run Stop Break Debug Options Help
assignment_2_1.cpp assignment_2_2.cpp matrix_trans.c matrix_addition.c nearest_point.c M4I320.c
1 #include <stdio.h>
2 #include <math.h>
3
4 struct Point {
5     double x;
6     double y;
7 };
8
9 double distance(struct Point p1, struct Point p2) {
10     return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y, 2));
11 }
12
13 int main() {
14     int n;
15     scanf("%d", &n);
16     struct Point points[100];
17     for(int i = 0; i < n; i++) {
18         scanf("%lf %lf", &points[i].x, &points[i].y);
19     }
20     double min_dist = 1000000000;
resources Compile Log Find Results Close
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\OneDrive\桌面\C++\Structure\nearest_point.exe
- Output Size: 155.8447265625 KB
- Compilation Time: 0.22s
21
22
23     for(int i = 0; i < n; i++) {
24         for(int j = i + 1; j < n; j++) {
25             double dist = distance(points[i], points[j]);
26             if(dist < min_dist) {
27                 min_dist = dist;
28                 point1 = i;
29                 point2 = j;
30             }
31         }
32     }
33     printf("%d %d %.2lf\n", point1 + 1, point2 + 1, min_dist);
34
35
36     return 0;
37 }
```

```
assignment_2_1.cpp assignment_2_2.cpp matrix_trans.c matrix_addition.c nearest_point.c M4I320.c
18     scanf("%lf %lf", &points[i].x, &points[i].y);
19 }
20 double min_dist = 100000000;
int point1 = 0, point2 = 0;
21
22
23     for(int i = 0; i < n; i++) {
24         for(int j = i + 1; j < n; j++) {
25             double dist = distance(points[i], points[j]);
26             if(dist < min_dist) {
27                 min_dist = dist;
28                 point1 = i;
29                 point2 = j;
30             }
31         }
32     }
33     printf("%d %d %.2lf\n", point1 + 1, point2 + 1, min_dist);
34
35
36     return 0;
37 }
```

5. [MH320] ในวันปีใหม่ สนามกีฬาแห่งหนึ่งได้ประดับไฟที่พื้นสนามฟุตบอลเพื่อความสวยงาม ในการประดับไฟนั้นทำโดยแบ่งสนามสีเหลี่ยมเป็นช่องอยู่ๆ จำนวน N แฉะ M คอลัมน์ รวม  $N \times M$  ช่อง เจ้าของสนามได้เปิดสนามให้ประชาชนทั่วไปเข้าชมเพื่อความสวยงาม

อย่างไรก็ตาม โลกนี้ไม่มีอะไรหรือฟรี เจ้าของสนามจะต้องจ่ายค่าไฟให้กับไฟประดับเหล่านี้ เนื่องจากมีการประดับไฟเป็นລາຍລາຍຕ່າງໆค่าไฟของไฟแต่ละช่องไม่จำเป็นต้องเท่ากัน

เพื่อไม่ให้เป็นการขาดทุน เจ้าของสนามจึงได้จัดเครื่องไอพ่นเจ็ตส่วนบุคคลให้กับประชาชนเช่าเพื่อบินดูไฟประดับ เครื่องพ่นเจ็ตแต่ละเครื่องเมื่อเช่าไปแล้วจะผู้ใช้สามารถบินได้ทั้งสิ้น K ครั้ง ในการบินแต่ละครั้ง จะใช้เชื้อเพลิงมูลค่าเท่ากับ L บาท ดังนั้น ค่าใช้จ่ายรวมทั้งหมดของเจ้าของสนามคือค่าไฟรวมของไฟประดับ และค่าเชื้อเพลิงรวมของการบินเครื่องไอพ่นเจ็ตในการบินทั้งหมด

เจ้าของสนามทราบว่าจะมีคนมาชมและเช่าเครื่องไอพ่นเจ็ตจำนวน C คน เข้าต้องการคำนวณค่าเช่าเครื่องไอพ่นเจ็ตต่อคนที่น้อยที่สุด ที่จะทำให้เขามีขาดทุน เพื่อให้การเช่าเป็นไปได้สะดวก ค่าเช่าจะต้องเป็นจำนวนเต็มเสมอด้วย

เขียนโปรแกรมรับราคาค่าไฟ ของสนามแต่ละช่อง รวมทั้งข้อมูลของการใช้เครื่องไอพ่นเจ็ต จากนั้น คำนวณหาค่าเช่าเครื่องไอพ่นเจ็ตต่อคนที่เป็นจำนวนเต็มที่น้อยที่สุด ที่จะทำให้เจ้าของสนามไม่มีขาดทุน

### ข้อมูลอินพุต

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

บรรทัดที่สองระบุจำนวนเต็มบวก L และ K คันด้วยช่องว่าง โดยที่ L แทนราคาเชื้อเพลิงต่อการบินหนึ่งครั้ง และ K แทนจำนวนครั้งที่เครื่องไอพ่นใช้บินได้ต่อคนเช่าหนึ่งคน ( $1 \leq L \leq 100; 1 \leq K \leq 100$ )

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

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

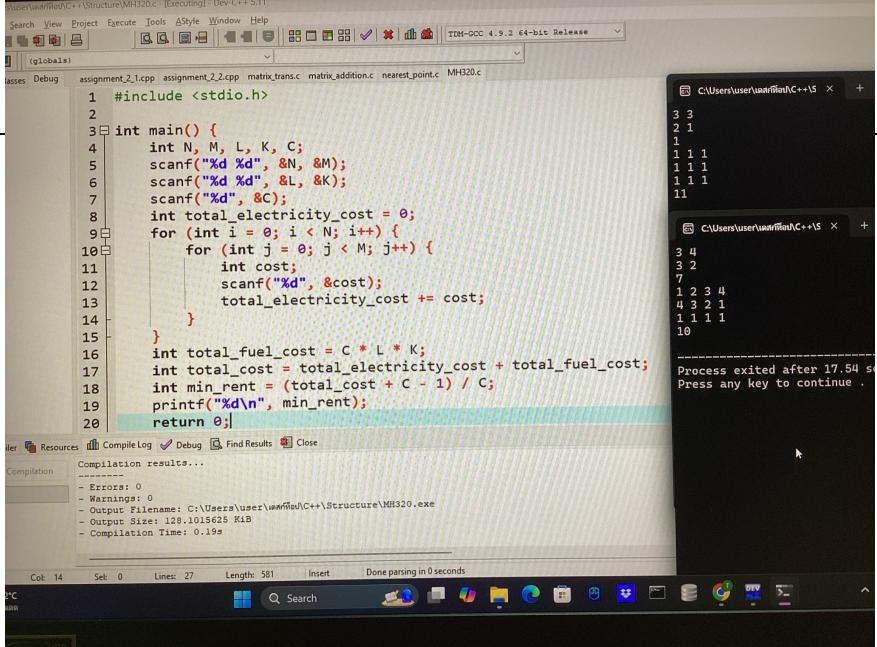
ชื่อ-นามสกุล..... **ກວດກັບທີ່** ລ້າທອງ..... รหัสประจำตัวนักศึกษา..... **6604062636208** .....

วันที่..... **10** เดือน..... **ມັງກອນ** ..... พ.ศ. 2562 Lab..... **6** ..... หมายเลขเครื่อง..... **—**

### ຂໍ້ມູນເລາທິພຸຖ

ມີປຽບທັດເດືອຍເປັນຈຳນວນເຕີມບາກທີ່ຈຳນວນ ແທນຄ່າເຊົ່າເຄື່ອງໄວໝ່າງເຈື້ອຕ່ອງການທີ່ເປັນຈຳນວນເຕີມທີ່ນ້ອຍທີ່ສຸດ ທີ່ຈະທຳໃຫ້ເຈົ້າຂອງສນາມໄນ່ຂາດທຸນ

### ຕັວອຢ່າງ

ອິນພຸຖ	ເລາທິພຸຖ
3 3 2 1 1 1 1 1 1 1 1 1 1 1	11
3 4 3 2 7 1 2 3 4 4 3 2 1 1 1 1 1	10
 <pre> 1 #include &lt;stdio.h&gt; 2 3 int main() { 4     int N, M, L, K, C; 5     scanf("%d %d", &amp;N, &amp;M); 6     scanf("%d %d", &amp;L, &amp;K); 7     scanf("%d", &amp;C); 8     int total_electricity_cost = 0; 9     for (int i = 0; i &lt; N; i++) { 10         for (int j = 0; j &lt; M; j++) { 11             int cost; 12             scanf("%d", &amp;cost); 13             total_electricity_cost += cost; 14         } 15     } 16     int total_fuel_cost = C * L * K; 17     int total_cost = total_electricity_cost + total_fuel_cost; 18     int min_rent = (total_cost + C - 1) / C; 19     printf("%d\n", min_rent); 20     return 0; } </pre>	

6. [bigdata] ຈຶ່ງເຂົ້ານໂປຣແກຣມເພື່ອທາຜລຄຸນຂອງຈຳນວນທີ່ມີນາດໃໝ່ (ສິ່ງໃໝ່ກ່ຽນກວ່າທີ່ long int ຈະສາມາຮັບໄດ້)

ชื่อ-นามสกุล..... ก้องพัฒน์ ลีลาทราย ..... รหัสประจำตัวนักศึกษา..... ๖๖๐๔๐๖๒๖๓๖๒๐๘ .....

วันที่..... ๑๐ เดือน..... กันยายน พ.ศ. ๒๕๖๒

Lab..... ๖ ..... หมายเลขอร่อง..... —

### ข้อมูลนำเข้า

ในบรรทัดแรก ประกอบด้วยเลขจำนวนเต็มบวกของตัวถูกดำเนินการตัวแรก (A)

บรรทัดที่ 2 คือเลขจำนวนเต็มบวกของตัวถูกดำเนินการตัวที่สอง (B)

ตัวเลข A และ B เป็นตัวมีค่าอย่างมาก 100 หลัก และแต่ละหลักเป็นเลขอะไรมีได้

### limit

ชนิดข้อมูลใช้ได้เฉพาะ int และ char เท่านั้น

Library ที่ใช้ได้ ได้แก่ stdio.h, string.h เท่านั้น

### ข้อมูลส่งออก

แสดงผลผลลัพธ์ของการคูณ

### ตัวอย่างข้อมูลนำเข้า

### ตัวอย่างข้อมูลส่งออก

12	132
11	
123456789123456789123456789	987654312987654312987654312
8	
999999999	999999998000000001
999999999	
111222333444555666777888999	1112099630370988148518765544444
99988877766655544433222111	

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

void multiplyLargeNumbers(char num1[], char num2[], char result[]) {
    int len1 = strlen(num1), len2 = strlen(num2);
    int product[len1 + len2];
    memset(product, 0, sizeof(product));

    for (int i = len1 - 1; i >= 0; i--) {
        for (int j = len2 - 1; j >= 0; j--) {
            int mul = (num1[i] - '0') * (num2[j] - '0');
            int p1 = i + j, p2 = i + j + 1;
            int sum = mul + product[p2];

            product[p2] = sum % 10;
            product[p1] += sum / 10;
        }
    }

    int k = 0;
}

int main() {
    char num1[101], num2[101], result[202];
    scanf("%s", num1);
    scanf("%s", num2);
    multiplyLargeNumbers(num1, num2, result);
    printf("%s\n", result);

    return 0;
}
```

```
int k = 0;
while (k < len1 + len2 && product[k] == 0) k++;

int index = 0;
for (int k = len1 + len2; k >= 0; k++) {
    result[index++] = product[k] + '0';
}
result[index] = '\0';

if (strlen(result) == 0) {
    strcpy(result, "0");
}

int main() {
    char num1[101], num2[101], result[202];
    scanf("%s", num1);
    scanf("%s", num2);
    multiplyLargeNumbers(num1, num2, result);
    printf("%s\n", result);

    return 0;
}
```