



การทดลองที่ 5-2 ฟังก์ชัน ตอนที่ 2 Function and Array

วัตถุประสงค์

1. ศึกษาการทำงานของฟังก์ชัน (Function) ที่มีการรับพารามิเตอร์แบบอาร์เรย์
2. ฝึกการเรียกใช้งานฟังก์ชัน และส่งค่าอาร์เรย์ให้กับฟังก์ชันได้อย่างถูกต้อง
3. รู้จักการส่งผ่านค่าให้กับฟังก์ชันแบบ Pass-by-value และ Pass-by-reference

ตอนที่ 1 การส่งผ่านค่าบางอิลีเมนต์ และทุกอิลีเมนต์ของอาร์เรย์ให้กับฟังก์ชัน

1. ทดลองโปรแกรมตัวอย่าง

1.1 พิมพ์โปรแกรม lab5_2p1.c

```
1. #include <stdio.h>
2. void show1(int x);
3. int main()
4. { int i;
5.   int a[5]={3, -1, 2, 7, 8};
6.   printf("Show all elements of array\n");
7.   for( i=0; i<5; i++)
8.       show1(a[i]);
9.   printf("\nEnd of program.\n");
10.  getch();
11.  return 0;
12. }
13. void show1(int x)
14. {
15.     printf(" %d ", x);
16. }
```

- 1.2 จงแก้ไข/เพิ่มเติม โปรแกรม โดยสร้างฟังก์ชัน show2 ที่รับข้อมูลเป็นอาร์เรย์ และแสดงค่า ข้อมูลทุกอิลีเมนต์ของอาร์เรย์นั้น (n คือจำนวนอิลีเมนต์ของอาร์เรย์) และให้มีการเรียกใช้ฟังก์ชัน show2 จากฟังก์ชัน main

โปรโตไทป์ของฟังก์ชัน void show2(int x[],int n);



ตอนที่ 2 การแก้ไขค่าของอาร์เรย์ในฟังก์ชัน

2. ทดลองโปรแกรมตัวอย่าง ซึ่งเป็นโปรแกรม เปลี่ยนค่าตัวเลขให้เป็นค่าสัมบูรณ์ (absolute)

2.1 พิมพ์โปรแกรม lab5_2p2.c

```
1. #include <stdio.h>
2. void absolute(int x[5]);
3. int main()
4. { int i;
5.     int a[5]={3, -1, 0, -7, -8};
6.     printf("Show all elements of array\n");
7.     for( i=0; i<5; i++) printf("%d ", a[i]);
8.     absolute(a);
9.     printf("\nAfter calling function absolute\n");
10.    for( i=0; i<5; i++) printf("%d ", a[i]);
11.    getch();
12.    return 0;
13. }
14. void absolute(int x[5])
15. { int j;
16.     for(j=0; j<5; j++)
17.         if (x[j]<0) x[j] = -1*x[j];
18. }
```

2.2 ให้แก้ไข/เพิ่มเติม โปรแกรมตัวอย่างโดยเพิ่มฟังก์ชัน square ที่ทำการเปลี่ยนค่าของทุก
อีลีเมนต์ของอาร์เรย์ให้เป็นค่ายกกำลังสอง

โปรโตไทป์ของฟังก์ชัน **void square(int x[5]);**

Checkpoint 1 จงเขียนโปรแกรม รับค่าคะแนนและเกรดซึ่งมี 2 ค่าคือ P และ F (Pass/Fail)

โดยเกรด P จะต้องได้คะแนนมากกว่าหรือเท่ากับ 50.0 กำหนดให้โปรแกรม รับค่าจำนวนนักเรียน N
(ไม่เกิน 100 คน) แล้ววนรับค่าคะแนนของแต่ละคน เก็บไว้ในอาร์เรย์ score จากนั้น เรียกใช้
ฟังก์ชัน grading เพื่อคำนวณเกรด เก็บไว้ในอาร์เรย์ grade แล้วให้โปรแกรมรายงานค่าคะแนน
และเกรดของแต่ละคน หลังการเรียกใช้ฟังก์ชัน grading แล้ว

โปรโตไทป์ของฟังก์ชัน

void grading(int N, float score[100], char grade[100]);



ตัวอย่างการรันโปรแกรม

```
Enter number of students N: 5
Score of student 1: 49.1
Score of student 2: 55
Score of student 3: 90.3
Score of student 4: 68.7
Score of student 5: 74
```

```
-----
Grade Report   Score   Grade
Student 1      49.1     F
Student 2      55.0     P
Student 3      90.3     P
Student 4      68.7     P
Student 5      74.0     P
```

Checkpoint 2 จงเขียนโปรแกรม รับค่าเมตริกซ์ขนาด 3x2 แล้วหาค่าที่มากที่สุด และ น้อยที่สุด

ในเมตริกซ์นั้น กำหนดให้มีการสร้างฟังก์ชัน min และ max สำหรับหาค่าต่ำสุดและสูงสุด และ

ฟังก์ชันสำหรับการอินพุตค่าของเมตริกซ์

โปรดใส่ชื่อของฟังก์ชัน

```
void inputMatrix(int M[3][2]);
int min(int M[3][2]);
int max(int M[3][2]);
```

ตัวอย่างการรันโปรแกรม

```
Enter elements of Matrix M[3x2]
```

```
M[1][1]: 201
```

```
M[1][2]: 200
```

```
M[2][1]: 0
```

```
M[2][2]: -79
```

```
M[3][1]: 580
```

```
M[3][2]: 200
```

```
Matrix
```

```
201 200
```

```
0 -79
```

```
580 200
```

```
Min = -79
```

```
Max = 580
```