

# แบบฝึกหัดปฏิบัติการควบคุมที่ 7: Function

ชื่อ-นามสกุล.....

ห้องที่ศูนย์ ลำดับ

รหัสประจำตัวนักศึกษา..... ๖๖๐๔๐๖๒๖๙๖๒๐๓

วันที่..... ๗.....เดือน..... กุมภาพันธ์..... พ.ศ. 2566

Section..... ๔ .....

## [การเรียนใช้ฟังก์ชันมาตรฐาน]

1. ให้แสดงค่าของ  $x$  หลังจากการใช้คำสั่งต่อไปนี้

a)  $x = \text{fabs}( 7.5 );$

7.5

b)  $x = \text{floor}( 7.5 );$

7.0

c)  $x = \text{fabs}( 0.0 );$

0.0

d)  $x = \text{ceil}( 0.0 );$

0.0

e)  $x = \text{fabs}( -6.4 );$

6.4

f)  $x = \text{ceil}( -6.4 );$

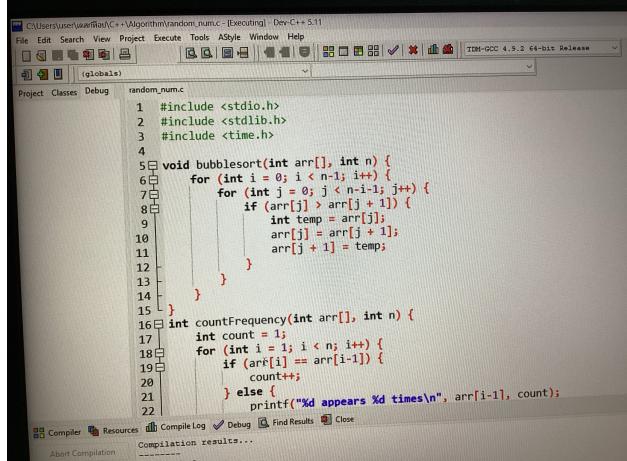
-6.0

g)  $x = \text{ceil}( -\text{fabs}( -8 + \text{floor}( -5.5 ) ) );$

-14.0

2. จงเขียนโปรแกรมเพื่อสุ่มเลขที่อยู่ระหว่าง 0-1000 ด้วยฟังก์ชัน `rand()` เก็บใส่ในอาร์เรย์จำนวน  $N$  ค่าเรียงข้อมูลที่ได้แล้วนับจำนวนความถี่ของเลขแต่ละค่า โดยใช้ฟังก์ชัน `printf` พร้อมแสดงผลลัพธ์

```
#include <stdio.h>
void main()
{
```

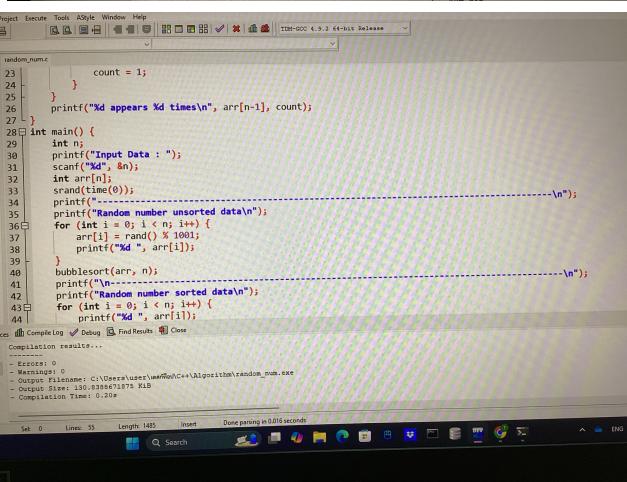


```

    #include <stdio.h>
    #include <stdlib.h>
    #include <time.h>
    ...
    void bubblesort(int arr[], int n) {
        for (int i = 0; i < n-1; i++) {
            for (int j = 0; j < n-i-1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
    int countFrequency(int arr[], int n) {
        int count = 1;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i-1]) {
                count++;
            } else {
                printf("%d appears %d times\n", arr[i-1], count);
                count = 1;
            }
        }
    }
}

int main() {
    int n;
    printf("Input Data : ");
    scanf("%d", &n);
    int arr[n];
    srand((unsigned)time(0));
    printf("-----\n");
    printf("Random number unsorted data\n");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 1001;
        printf("%d ", arr[i]);
    }
    bubblesort(arr, n);
    printf("\n");
    printf("Random number sorted data\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}

```

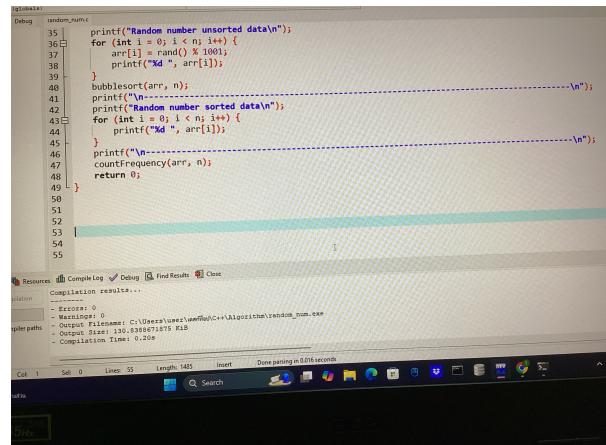


```

    #include <stdio.h>
    #include <stdlib.h>
    #include <time.h>
    ...
    void bubblesort(int arr[], int n) {
        for (int i = 0; i < n-1; i++) {
            for (int j = 0; j < n-i-1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
    int countFrequency(int arr[], int n) {
        int count = 1;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i-1]) {
                count++;
            } else {
                printf("%d appears %d times\n", arr[i-1], count);
                count = 1;
            }
        }
    }
}

int main() {
    int n;
    printf("Input Data : ");
    scanf("%d", &n);
    int arr[n];
    srand((unsigned)time(0));
    printf("-----\n");
    printf("Random number unsorted data\n");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 1001;
        printf("%d ", arr[i]);
    }
    bubblesort(arr, n);
    printf("\n");
    printf("Random number sorted data\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}

```

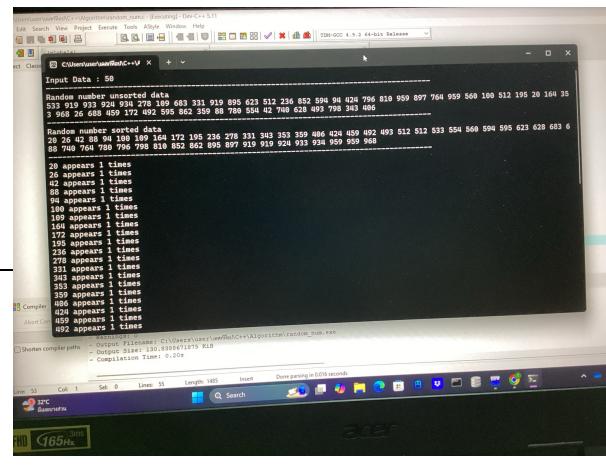


```

    #include <stdio.h>
    #include <stdlib.h>
    #include <time.h>
    ...
    void bubblesort(int arr[], int n) {
        for (int i = 0; i < n-1; i++) {
            for (int j = 0; j < n-i-1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
    int countFrequency(int arr[], int n) {
        int count = 1;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i-1]) {
                count++;
            } else {
                printf("%d appears %d times\n", arr[i-1], count);
                count = 1;
            }
        }
    }
}

int main() {
    int n;
    printf("Input Data : ");
    scanf("%d", &n);
    int arr[n];
    srand((unsigned)time(0));
    printf("-----\n");
    printf("Random number unsorted data\n");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 1001;
        printf("%d ", arr[i]);
    }
    bubblesort(arr, n);
    printf("\n");
    printf("Random number sorted data\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}

```



```

    #include <stdio.h>
    #include <stdlib.h>
    #include <time.h>
    ...
    void bubblesort(int arr[], int n) {
        for (int i = 0; i < n-1; i++) {
            for (int j = 0; j < n-i-1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
    int countFrequency(int arr[], int n) {
        int count = 1;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i-1]) {
                count++;
            } else {
                printf("%d appears %d times\n", arr[i-1], count);
                count = 1;
            }
        }
    }
}

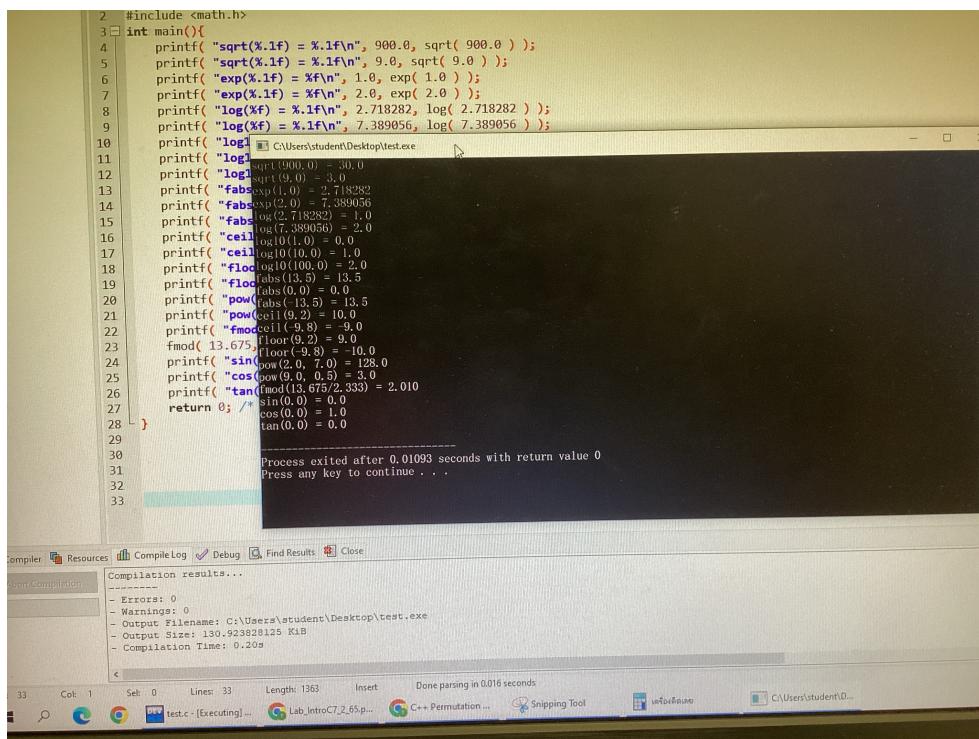
int main() {
    int n;
    printf("Input Data : ");
    scanf("%d", &n);
    int arr[n];
    srand((unsigned)time(0));
    printf("-----\n");
    printf("Random number unsorted data\n");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 1001;
        printf("%d ", arr[i]);
    }
    bubblesort(arr, n);
    printf("\n");
    printf("Random number sorted data\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}

```

## 3. จากโปรแกรมต่อไปนี้ให้แสดงผลลัพธ์ของโปรแกรมจากการใช้ฟังก์ชันมาตราฐานต่าง ๆ

```

1 #include <stdio.h>
2 #include <math.h>
3 int main( void )
4 {
5     printf( "sqrt(%.1f) = %.1f\n", 900.0, sqrt( 900.0 ) );
6     printf( "sqrt(%.1f) = %.1f\n", 9.0, sqrt( 9.0 ) );
7     printf( "exp(%.1f) = %f\n", 1.0, exp( 1.0 ) );
8     printf( "exp(%.1f) = %f\n", 2.0, exp( 2.0 ) );
9     printf( "log(%f) = %.1f\n", 2.718282, log( 2.718282 ) );
10    printf( "log(%f) = %.1f\n", 7.389056, log( 7.389056 ) );
11    printf( "log10(%.1f) = %.1f\n", 1.0, log10( 1.0 ) );
12    printf( "log10(%.1f) = %.1f\n", 10.0, log10( 10.0 ) );
13    printf( "log10(%.1f) = %.1f\n", 100.0, log10( 100.0 ) );
14    printf( "fabs(%.1f) = %.1f\n", 13.5, fabs( 13.5 ) );
15    printf( "fabs(%.1f) = %.1f\n", 0.0, fabs( 0.0 ) );
16    printf( "fabs(%.1f) = %.1f\n", -13.5, fabs( -13.5 ) );
17    printf( "ceil(%.1f) = %.1f\n", 9.2, ceil( 9.2 ) );
18    printf( "ceil(%.1f) = %.1f\n", -9.8, ceil( -9.8 ) );
19    printf( "floor(%.1f) = %.1f\n", 9.2, floor( 9.2 ) );
20    printf( "floor(%.1f) = %.1f\n", -9.8, floor( -9.8 ) );
21    printf( "pow(%.1f, %.1f) = %.1f\n", 2.0, 7.0, pow( 2.0, 7.0 ) );
22    printf( "pow(%.1f, %.1f) = %.1f\n", 9.0, 0.5, pow( 9.0, 0.5 ) );
23    printf( "fmod(%.3f/%.3f) = %.3f\n", 13.675, 2.333,
24        fmod( 13.675, 2.333 ) );
25    printf( "sin(%.1f) = %.1f\n", 0.0, sin( 0.0 ) );
26    printf( "cos(%.1f) = %.1f\n", 0.0, cos( 0.0 ) );
27    printf( "tan(%.1f) = %.1f\n", 0.0, tan( 0.0 ) );
28    return 0; /* indicates successful termination */
29 }
```



## [ພັບປຸງກຳນົດເອງ]

1. ຈົນເຂີຍໂປຣແກຣມຫາຄ່າ  $f(x)$  ໂດຍສາມາດ  $f(x)$  ເປັນດັ່ງນີ້

$$\begin{aligned} f(x) &= x^2 + 2x + 3 \quad \text{if } x < 0 \\ &= 0 \quad \quad \quad \text{if } x = 0 \\ &= x - 2 \quad \quad \quad \text{if } x > 0 \end{aligned}$$

ກຳນົດໃຫ້ສ່ວນທີ່ໃຊ້ໃນການຄຳນວນຄ່າ  $f(x)$  ອູ້ຢູ່ໃນພັບປຸງ  $fx$  ກຳນົດໃຫ້ສ່ວນທີ່ຮັບຄ່າຕົວແປ່ງ  $x$  ຈາກຄືບອິນໄຕ ແລະ ສ່ວນທີ່ແສດງຜລລັບອິນໄຕຂອງຄ່າ  $f(x)$  ອູ້ຢູ່ໃນພັບປຸງ  $main$  ມ້ານໃຊ້ຕົວແປ່ງ  $Global$  ໃນໂປຣແກຣມເຕີດຂາດ

```
#include <stdio.h>
void main()
{
```

```

1 #include <stdio.h>
2
3 int FX(int x);
4
5 int main(){
6     int x;
7     printf("f(x) = x^2 + 2x + 3 if x < 0\n");
8     printf("f(x) = 0 if x = 0\n");
9     printf("f(x) = x - 2 if x > 0\n");
10    printf("\nInput x : ");
11    scanf("%d", &x);
12    int result = FX(x);
13    printf("result : %d", result);
14    return 0;
15 }
16
17 int FX(int x){
18     if(x < 0){
19         return x*x + 2*x + 3;
20     } else if(x == 0){
21         return 0;
22     } else if(x > 0){
23         return x - 2;
24     }
25 }
```

```

9 |     printf("f(x) = x - 2 if x > 0\n");
10|     printf("\nInput x : ");
11|     scanf("%d", &x);
12|     int result = FX(x);
13|     printf("result : %d", result);
14|     return 0;
15 }
16
17 int FX(int x){
18     if(x < 0){
19         return x*x + 2*x + 3;
20     } else if(x == 0){
21         return 0;
22     } else if(x > 0){
23         return x - 2;
24     }
25 }
```

C:\Users\student\Desktop\test.c

f(x) = x^2 + 2x + 3 if x < 0  
f(x) = 0 if x = 0  
f(x) = x - 2 if x > 0

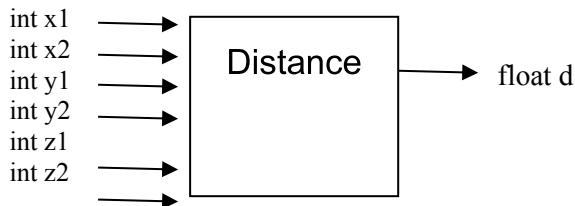
Input x : -1  
result : 2

Process exited after 0.559 seconds with return value 0  
Press any key to continue . . .

2. จงเขียนโปรแกรมเพื่อคำนวณระยะห่างระหว่างจุดสองจุด คือ  $(x_1, y_1, z_1)$  และ  $(x_2, y_2, z_2)$  โดยการใช้ฟังก์ชัน Distance โดยฟังก์ชันจะรับ input เป็นเลขจำนวนเต็ม 6 ตัว สำหรับค่า  $x_1, y_1, z_1, x_2$  และ  $y_2, z_2$  จากนั้นจะคำนวณสี่เหลี่ยมลับค่าตัวเลขทศนิยมสองหลักเป็นค่าระยะห่าง (distance)

$$d = \sqrt{|z_2-z_1|^2 + |y_2-y_1|^2 + |x_2-x_1|^2}$$

$$d = \sqrt{|z_2-z_1|^2 + |y_2-y_1|^2 + |x_2-x_1|^2}$$



```
#include <stdio.h>
void main()
{
    float distance(float x1, float y1, float z1, float x2, float y2, float z2){
        return sqrt(pow(z2-z1, 2) + pow(y2-y1, 2) + pow(x2-x1, 2));
    }
    int main(){
        float x1, y1, z1, x2, y2, z2;
        scanf("%f %f %f %f %f", &x1, &y1, &z1, &x2, &y2, &z2);
        printf("%.2f\n", distance(x1, y1, z1, x2, y2, z2));
        return 0;
    }
}
```

**Output 1:**

**Output 2:**

## 3. จงเขียนโปรแกรมเพื่ocomputingพื้นที่สี่เหลี่ยมและวงกลม ดังนี้

ในส่วนของโปรแกรมหลัก ให้แสดงเมนูให้ผู้ใช้เลือกว่าต้องการคำนวณพื้นที่ของสี่เหลี่ยมหรือวงกลม

โดยถ้าผู้ใช้เลือกเมนู 1 ให้รับค่าความกว้างและความยาว (กำหนดให้ความกว้างและความยาวเป็นจำนวนเต็ม) และเรียกใช้ฟังก์ชันในการคำนวณสี่เหลี่ยม

แต่ถ้าผู้ใช้เลือกเมนูข้อ 2 ให้รับค่ารัศมี (เป็นทศนิยม) และเรียกใช้ฟังก์ชันในการคำนวณพื้นที่วงกลม

เมื่อเรียกใช้งานฟังก์ชันดังกล่าวเพื่อคำนวณค่าพื้นที่แล้ว ให้พิงก์ชันนั้นๆ ส่งค่าผลลัพธ์กลับมายังโปรแกรมหลัก และแสดงผลลัพธ์ออกทางหน้าจอ

ตัวอย่างการทำงานของโปรแกรม

===== MENU =====

1. Calculate area of rectangle
2. Calculate area of circle

Please enter 1 or 2: 1

Please enter width: 5

Please enter height: 10

Area = 50.00

หรือ

===== MENU =====

1. Calculate area of rectangle
2. Calculate area of circle

Please enter 1 or 2: 2

Please enter radius: 10.5

Area = 346.36

The screenshot shows three panels of a C++ IDE (Dev C++ 5.11) displaying code and terminal output.

- Panel 1:** Shows the main function of a program that calculates the area of a rectangle or circle based on user input. It includes prompts for width/height or radius, and prints the result to the terminal.
- Panel 2:** Shows the implementation of the rectangle area calculation function (`float calculaterectangle(float width, float height)`) and the circle area calculation function (`float calculatecircle(float radius, float pi)`).
- Panel 3:** Shows the terminal output of the program running. It asks for a choice (1 or 2), then for width and height (or radius), and finally prints the calculated area (50.00 or 346.36) followed by a return value of 0.

```

1 //include <stdio.h>
2
3 float calculaterectangle(float width, float height);
4 float calculatecircle(float radius, float pi);
5
6 int main(){
7     int choice;
8     float result1 = 0, result2 = 0;
9     float pi = 3.141592654, width, height, radius;
10    printf("===== MENU =====\n");
11    printf("1. Calculate area of rectangle\n");
12    printf("2. Calculate area of circle\n");
13    printf("\nPlease enter 1 or 2: ");
14    scanf("%d", &choice);
15    if(choice == 1){
16        printf("Please enter width: ");
17        scanf("%f", &width);
18        printf("Please enter height: ");
19        scanf("%f", &height);
20        result1 = calculaterectangle(width, height);
21        printf("\nArea = %.2f\n", result1);
22    } else if(choice == 2){
23        printf("Please enter radius: ");
24        scanf("%f", &radius);
25        result2 = calculatecircle(radius, pi);
26        printf("\nArea = %.2f\n", result2);
27    }
28    return 0;
29 }
30
31 float calculaterectangle(float width, float height){
32     return width * height;
33 }
34
35 float calculatecircle(float radius, float pi){
36     return pi * radius * radius;
37 }
38
39
40
=====
MENU
1. Calculate area of rectangle
2. Calculate area of circle
Please enter 1 or 2: 1
Please enter width: 5
Please enter height: 10
Area = 50.00
=====
MENU
1. Calculate area of rectangle
2. Calculate area of circle
Please enter 1 or 2: 2
Please enter radius: 10.5
Area = 346.36
Process exited after 7.14 seconds with return value 0
Press any key to continue . .

```

4. จากโปรแกรมต่อไปนี้ จงเติมค่าตัวแปรลงในช่องว่างที่กำหนดให้ (อธิบายการทำงานของโปรแกรมในช่วงไม่ Lab)

<pre> /* 1   #include &lt;stdio.h&gt; */ /* 2   /* function prototype */ /* 3   int f1( int ); /* 4   float f2( int *, int ); */ /* 5   /* global variable */ /* 6   int a = 10; /* 7   int b = 5; */ /* 8   int f1( int a ) { /* 9   { /* 10   return a - 1; */ /* 11   float f2( int *k, int y ) { /* 12   *k = f1(a); /* 13   a *= 5; /* 14   y = f1(a); /* 15   return *k + y + 0.5; */ /* 16   } */ /* 17   int main() { /* 18   int x, y; /* 19   x = f1(a + b); /* 20   y = f1(b); */ /* 21   printf("%d\n", x); /* 22   printf("%.2f\n", f2(&amp;x, y)); /* 23   printf("%d\n", a); /* 24   printf("%d\n", x); /* 25   printf("%d\n", y); */ /* 26   } */ /* 27   } </pre>	<hr/> <b>14</b> <hr/> <hr/> <b>58.50</b> <hr/> <hr/> <b>50</b> <hr/> <hr/> <b>9</b> <hr/> <hr/> <b>4</b> <hr/>
--	--

ชื่อ-นามสกุล..... กองกัณณ์ คำคง ..... รหัสประจำตัวนักศึกษา..... ๖๖๐๔๐๖๒๖๓๖๒๐๘ .....

วันที่..... ๗.....เดือน..... กุมภาพันธ์ ..... พ.ศ. ๒๕๖๖ ..... ห้องเรียน Lab ที่..... ๔ .....

/* 28 */ /* 29 */ /* 30 */ /* 31 */ /* 32 */ /* 33 */ /* 34 */ /* 35 */ /* 36 */ /* 37 */ /* 38 */ /* 39 */		
--	--	--

## 5. จงเขียนฟังก์ชันเพื่อหาค่าต่ำสุด (minArray(int A[])) ค่าสูงสุด(maxArray(int A[]))

ผลรวมของตัวเลข(sumArray(int A[])) ที่อยู่ในอาร์เรย์

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

```
void main()
```

```
{
```

```
random_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c
1 #include <stdio.h>
2
3 int minArray(int arr[], int n);
4 int maxArray(int arr[], int n);
5 int sumArray(int arr[], int n);
6
7 int main(){
8     int n;
9     scanf("%d", &n);
10    int arr[n];
11    for(int i=0; i<n; i++){
12        scanf("%d", &arr[i]);
13    }
14    int result1 = minArray(arr, n);
15    printf("minArray is : %d\n", result1);
16    int result2 = maxArray(arr, n);
17    printf("maxArray is : %d\n", result2);
18    int summar = sumArray(arr, n);
19    printf("sumArray is : %d\n", summar);
20
21    return 0;
22}
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\Downloads\Structure\mixmaxarr.exe
- Output Size: 129.154296875 Kib
- Compilation Time: 0.20s

Done parsing in 0 seconds

```
random_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c
33 int maxArray(int arr[], int n){
34     int max = arr[0];
35     for(int i=0; i<n; i++){
36         if(arr[i] > max){
37             max = arr[i];
38         }
39     }
40     return max;
41 }
```

```
C:\Users\user\Downloads\Structure\mixmaxarr.exe
5 2 8 3 1
minArray is : 1
maxArray is : 8
sumArray is : 19
Process exited after 5.881 seconds with return value 0
Press any key to continue . . .
```

Project Classes Debug random\_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c

```
22
23 int minArray(int arr[], int n){
24     int min = arr[0];
25     for(int i=0; i<n; i++){
26         if(arr[i] < min){
27             min = arr[i];
28         }
29     }
30     return min;
31 }
32
33 int maxArray(int arr[], int n){
34     int max = arr[0];
35     for(int i=0; i<n; i++){
36         if(arr[i] > max){
37             max = arr[i];
38         }
39     }
40     return max;
41 }
42
43 int sumArray(int arr[], int n){
44     int sum = 0;
45     for(int i=0; i<n; i++){
46         sum+=arr[i];
47     }
48     return sum;
49 }
```

Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\Downloads\Structure\mixmaxarr.exe
- Output Size: 129.154296875 Kib
- Compilation Time: 0.20s

Done parsing in 0 seconds

Compiler Resources Compile Log Debug Find Results Close

```
Abort Compilation
Shorten compiler paths
random_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c
54
55 int minArray(int arr[], int n){
56     for(int i=0; i<n; i++){
57         if(arr[i] < min){
58             min = arr[i];
59         }
60     }
61     return min;
62 }
63
64 int maxArray(int arr[], int n){
65     int max = arr[0];
66     for(int i=0; i<n; i++){
67         if(arr[i] > max){
68             max = arr[i];
69         }
70     }
71     return max;
72 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\Downloads\Structure\mixmaxarr.exe
- Output Size: 129.154296875 Kib
- Compilation Time: 0.20s

Done parsing in 0 seconds

Compiler Resources Compile Log Debug Find Results Close

```
Abort Compilation
Shorten compiler paths
random_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c
54
55 int minArray(int arr[], int n){
56     for(int i=0; i<n; i++){
57         if(arr[i] < min){
58             min = arr[i];
59         }
60     }
61     return min;
62 }
63
64 int maxArray(int arr[], int n){
65     int max = arr[0];
66     for(int i=0; i<n; i++){
67         if(arr[i] > max){
68             max = arr[i];
69         }
70     }
71     return max;
72 }
73
74 int sumArray(int arr[], int n){
75     int sum = 0;
76     for(int i=0; i<n; i++){
77         sum+=arr[i];
78     }
79     return sum;
80 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\Downloads\Structure\mixmaxarr.exe
- Output Size: 129.154296875 Kib
- Compilation Time: 0.20s

Done parsing in 0 seconds

Compiler Resources Compile Log Debug Find Results Close

```
Abort Compilation
Shorten compiler paths
random_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c
51
52
53
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\Downloads\Structure\mixmaxarr.exe
- Output Size: 129.154296875 Kib
- Compilation Time: 0.20s

Done parsing in 0 seconds

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation
Shorten compiler paths
random\_num.c findDistance.c recAndcic.c lab7.c mixmaxarr.c

6. กำหนด function prototype สำหรับวาดเส้น และวาดจุดดังนี้

```
void drawline ( char c, int begin, int end );
```

ฟังก์ชัน drawline จะแสดงผลอักษร c จากตำแหน่ง begin จนถึงตำแหน่ง end

```
เช่น drawline('+', 5, 10); จะแสดงผล
```

```
+++++
```

```
void drawpoints ( char c, int *list, int n );
```

ฟังก์ชัน drawpoints จะแสดงผลอักษร c ในตำแหน่งที่ระบุใน list (เรียงลำดับจากซ้ายไปขวา) โดยที่ n คือจำนวนจุดที่ต้องการแสดงผลในบรรทัดนั้น

```
เช่น int points[ ] = {1, 3, 5};
```

```
drawpoint('$', points, 3); จะแสดงผล
```

```
$ $ $
```

จะเขียนโปรแกรมเพื่อวาดรูปสามเหลี่ยมและสี่เหลี่ยม (โดยเรียกใช้ Function ตั้งกล่าว) เพื่อให้แสดงผลดังนี้  
ตัวอย่างผลการรันโปรแกรม

<pre> * * * * *   * *****# #     # #     # #####</pre>	<p>กำหนดโปรแกรมหลักในการทดสอบ function drawline และ drawpoints ดังนี้</p> <pre> int main( ) {     int list[10];     char c = '*';      /* draw triangle */     list[0] = 5;     drawpoints(c, list, 1);     list[0] = 3; list[1] = 7;     drawpoints(c, list, 2);     list[0] = 1; list[1] = 9;     drawpoints(c, list, 2);     drawline(c, 1, 10);     /* draw rectangle */     printf("\n");     c = '#';     drawline(c, 1, 10);     list[0] = 1; list[1] = 10;     drawpoints(c, list, 2);     drawpoints(c, list, 2);     drawline(c, 1, 10);      getchar(); }</pre>
--	--

```
random_num.c findDistance.c recAndcir.c lab7.c mixmaxarr.c draw.c
```

```
1 #include <stdio.h>
2
3 void drawline(char c, int begin, int end) {
4     for (int i = begin; i <= end; i++) {
5         printf("%c", c);
6     }
7     printf("\n");
8 }
9 void drawpoints(char c, int *list, int n) {
10    for (int i = 1; i <= 10; i++) {
11        int found = 0;
12        for (int j = 0; j < n; j++) {
13            if (i == list[j]) {
14                printf("%c", c);
15                found = 1;
16                break;
17            }
18        }
19        if (!found) {
20            printf(" ");
21        }
22    }
}
Resources Compile Log Debug Find Results Close
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\桌面\C++\Structure\draw.exe
- Output Size: 129.115234375 KiB
- Compilation Time: 0.19s
Done parsing in 0.015 seconds
```

```
* *
*   *
*****
#####
#   #
#   #
#####
-----
```

Process exited after 0.36 seconds with r  
Press any key to continue . . .

```
(globals)
```

```
random_num.c findDistance.c recAndcir.c lab7.c mixmaxarr.c draw.c
```

```
25 int main()
26 {
27     int list[10];
28     char c = '*';
29     /* draw triangle */
30     list[0] = 5;
31     drawpoints(c, list, 1);
32     list[0] = 3; list[1] = 7;
33     drawpoints(c, list, 2);
34     list[0] = 1; list[1] = 9;
35     drawpoints(c, list, 2);
36     drawline(c, 1, 10);
37     /* draw rectangle */
38     printf("\n");
39     c = '#';
40     drawline(c, 1, 10);
41     list[0] = 1; list[1] = 10;
42     drawpoints(c, list, 2);
43     drawpoints(c, list, 2);
44     drawline(c, 1, 10);
45     getchar();
46 }
```

```
Compiler Resources Compile Log Debug Find Results Close
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\user\桌面\C++\Structure\draw.exe
- Output Size: 129.115234375 KiB
- Compilation Time: 0.19s
Done parsing in 0.015 seconds
```