

ใบงานการทดลองที่ 11

เรื่อง ฟังก์ชัน

1. จุดประสงค์ทั่วไป

- 2.1. รู้และเข้าใจหลักภาษาที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

- 2.1.29. บอกและอธิบายฟังก์ชัน
- 2.1.30. ผูกมัดและทดลองใช้ฟังก์ชัน
- 2.1.31. ออกแบบแนวทางการใช้ฟังก์ชันเพื่อให้ทำงานได้มีประสิทธิภาพมากขึ้น
- 2.1.32. แนะนำแนวทางการใช้ฟังก์ชันอย่างเป็นระบบ

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

- 4.1. จงบอกและอธิบายความหมายของ "ฟังก์ชัน"

ฟังก์ชัน คือ คำสั่งที่สั่งให้คอมพิวเตอร์ทำอะไรบางอย่างเมื่อได้รับข้อมูลเข้า (input) แล้วจะส่งผลลัพธ์ (output) ออกมา

- 4.2. จงระบุประโยชน์ของ "ฟังก์ชัน"

ในบทเรียนนี้ เราจะเห็นว่าฟังก์ชันช่วยให้เราสามารถนำโค้ดที่เราเขียนไว้มาใช้ซ้ำได้โดยไม่ต้องเขียนซ้ำๆ กัน

- 4.3. จงยกตัวอย่างการสร้างโครงสร้างของฟังก์ชัน พร้อมวาดรูปประกอบการอธิบาย
วาดรูปประกอบ

```
int main() {  
    // คำสั่ง  
    return 0;  
}
```

คำอธิบาย

โค้ดตัวอย่างนี้คือฟังก์ชันที่รับค่าจาก main() และส่งค่ากลับคืนไปให้ main() เรียกใช้ฟังก์ชันนี้เพื่อทำอะไรบางอย่าง
Function นี้จะส่งค่ากลับคืนมาในคำสั่ง return คือฟังก์ชันที่
ที่ส่งค่ากลับคืนมา

5. ลำดับขั้นการปฏิบัติงาน

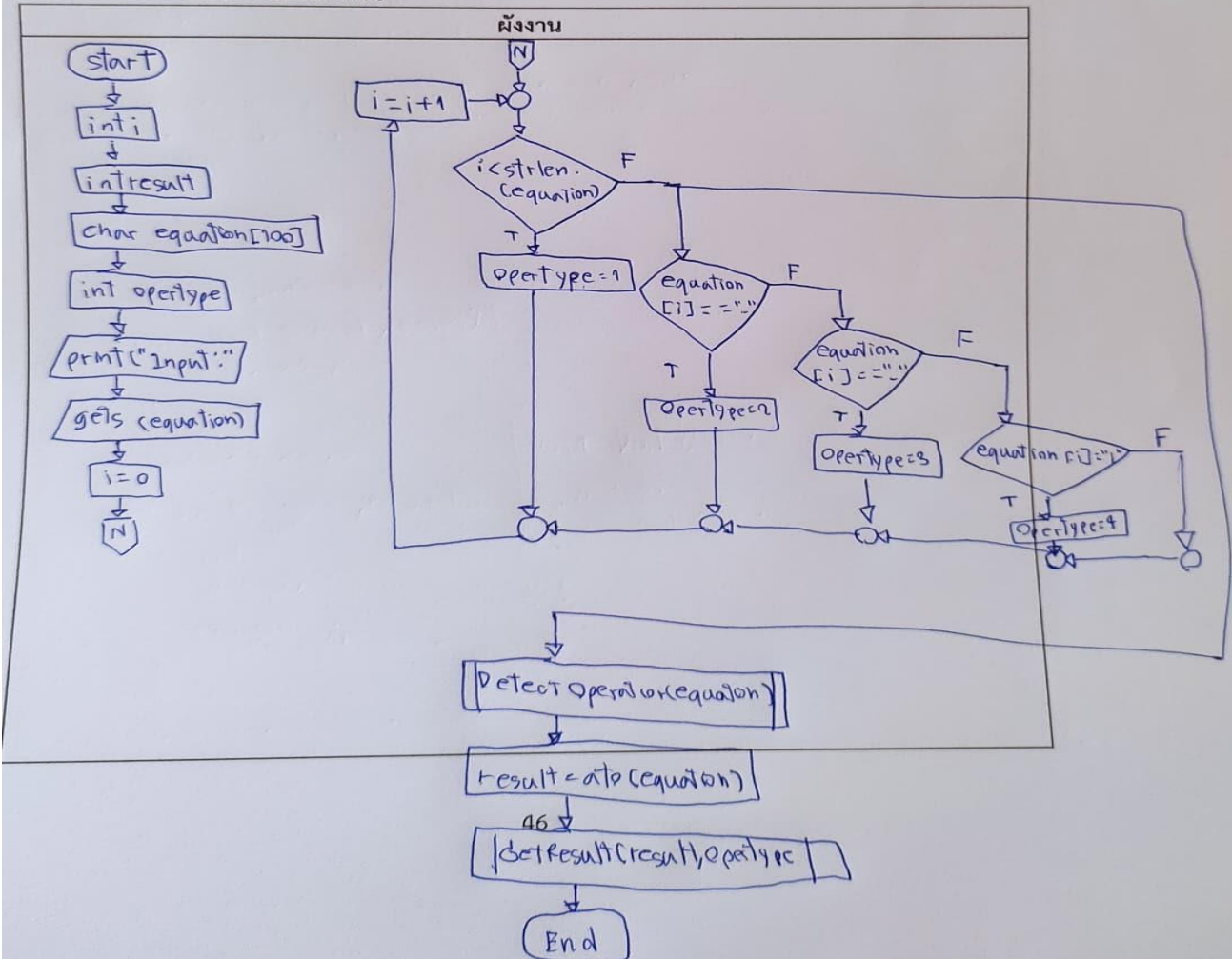
5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.1.1. จงเขียนโปรแกรมเครื่องคิดเลขอย่างง่าย ด้วยการรับชุดข้อความจากผู้ใช้ เพื่อนำมาหาผลลัพธ์ของคำตอบ โดยกำหนดให้ภายในฟังก์ชันหลักจะต้องมีการเรียกใช้ฟังก์ชันการทำงานย่อยเพื่อรับค่าจากผู้ใช้ ประมวลผลตัวดำเนินการ และแสดงผลลัพธ์ ดังแสดงฟังก์ชันต้นแบบดังนี้

```
char DetectOperator( char [] );
int GetResult( int, int );
```

	Test case 1	Test case 2
Input	Input : 3 * 4	Input : 5 - 9
Output	Result : 12	Result : -4

5.1.2. จงเขียนผังงาน



5.1.3. จงเขียนโค้ดโปรแกรม

โค้ดโปรแกรม

```

1  #include<stdio.h>
2  #include<string.h>
3  #include<cstdlib>
4
5  char    DetectOperator( char [] ) ;
6  int     GetResult ( int , int ) ;
7
8  int main() {
9      int    i ;
10     int    result ;
11     char    equation[100] ;
12     int     opertype ;
13     printf( "Input : " ) ;
14     gets( equation ) ;
15     for( i = 0 ; i < strlen( equation ) ; i++ ) {
16         if( equation[i] == '+' ){
17             |   opertype = 1 ;
18         }
19         else if ( equation[i] == '-' ){
20             |   opertype = 2 ;
21         }
22
23         else if ( equation[i] == '*' ){
24             |   opertype = 3 ;
25         }
26         else if ( equation[i] == '/' ){
27             |   opertype = 4 ;
28         }
29     }
30     DetectOperator(equation) ;
31     result = atoi(equation) ;
32     GetResult(result, opertype) ;
33     return 0 ;
34 }
35
36 char DetectOperator( char inoutput [] ) {
37     int i, j, k, n ;
38     int index, indexmove ;
39     int posimove = 0 ;
40     char strResult[100] ;
41     char operat, strnumber[10][100] , holder[100] ;
42     int num[100], result;
43     for( i = 0 ; i < strlen( inoutput ) ; i++ ) {

```

```

43     if( inoutput[i] == '+' || inoutput[i] == '-' || inoutput[i] == '*' || inoutput[i] == '/' ) {
44         index = i ;
45         operat = inoutput[i] ;
46     }
47 }
48 for ( j = 0 ; j <= 1 ; j++ ) {
49     indexmove = 0 ;
50     for ( k = posimove ; k < index ; k++ ) {
51         strnumber[j][indexmove] = inoutput[k] ;
52         holder[indexmove] = strnumber[j][indexmove] ;
53         indexmove++;
54     }
55     holder[indexmove] = '\0' ;
56     posimove = posimove + index + 1 ;
57     index = strlen( inoutput ) ;
58     num[j] = atoi(holder) ;
59 }
60
61 if ( operat == '+' ){
62     result = num[0] + num[1] ;
63     itoa(result, strResult, 10) ;

```

```

64     for ( n = 0 ; n < strlen(strResult) ; n++ ) {
65         inoutput[n] = strResult[n];
66     }
67     inoutput[n] = '\0' ;
68 }
69 else if ( operat == '-' ){
70     result = num[0] - num[1] ;
71     itoa(result, strResult, 10) ;
72     for ( n = 0 ; n < strlen(strResult) ; n++ ) {
73         inoutput[n] = strResult[n];
74     }
75     inoutput[n] = '\0' ;
76 }
77 else if ( operat == '*' ){
78     result = num[0] * num[1] ;
79     itoa(result, strResult, 10) ;
80     for ( n = 0 ; n < strlen(strResult) ; n++ ) {
81         inoutput[n] = strResult[n];
82     }
83     inoutput[n] = '\0' ;
84 }

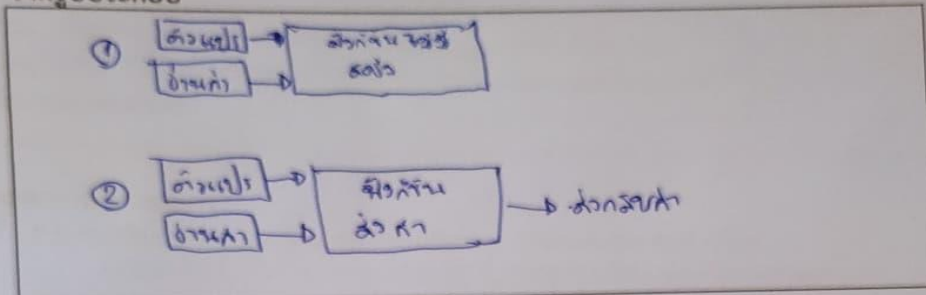
```

```
85     else if ( operat == '/' ){
86         result = num[0] / num[1] ;
87         itoa(result, strResult, 10) ;
88         for ( n = 0 ; n < strlen(strResult) ; n++ ) {
89             inoutput[n] = strResult[n];
90         }
91         inoutput[n] = '\0' ;
92     }
93 }
94 int GetResult( int number, int operat ){
95     printf( "Result" ) ;
96     if (operat == 1) {
97         printf(" of addition : %d ", number) ;
98     }
99     else if (operat == 2) {
100         printf(" of subtraction : %d ", number) ;
101     }
102     else if (operat == 3) {
103         printf(" of multiplication : %d ", number) ;
104     }
105     else if (operat == 4) {
```

```
106         printf(" of division : %d ", number) ;
107     }
108     return 0 ;
109 }
```


4.4. ฟังก์ชันที่มีการส่งกลับค่า และไม่มีการส่งกลับค่าแตกต่างกันอย่างไร อธิบายพร้อมวาดรูปประกอบคำอธิบาย และเขียนโค้ดตัวอย่างประกอบการอธิบาย

วาดรูปประกอบ



คำอธิบาย

ส่วนที่ 1 คือฟังก์ชัน ที่ส่งค่าส่งกลับแล้วส่งค่ากลับให้โปรแกรมเมอร์
ส่วนที่ 2 คือฟังก์ชัน ที่ส่งค่าส่งกลับแล้วส่งค่ากลับให้โปรแกรมเมอร์

ส่วนที่ 2 คือฟังก์ชัน ส่งค่าส่งกลับ แล้วส่งค่าส่งกลับให้โปรแกรมเมอร์
main สามารถ ส่งค่าส่งกลับ func. ส่งค่าส่งกลับให้โปรแกรมเมอร์

โค้ดโปรแกรมตัวอย่างประกอบคำอธิบาย

① ฟังก์ชันไม่ส่งค่าส่งกลับ

```

void Doit (Param, Param) {
    // คำสั่ง ;
}

```

② ฟังก์ชันส่งค่าส่งกลับ

```

int Do it (Param, Param) {
    int a
    // คำสั่ง ;
    return a;
}

```

```

void Doit (Param, Param) {
    // คำสั่ง
    Param = a; // ส่งค่าให้ parameter
}

```

6. สรุปผลการปฏิบัติงาน

จากการปฏิบัติงานพบว่าฟังก์ชัน atoi ทำหน้าที่แปลงค่า string เป็น int. ex. แปลงค่าฟังก์ชัน atoi จากแปลงจาก int เป็น string เพื่อใช้ในการคำนวณ string. ในตอนจบของข้อสอบจะได้ค่าเป็นค่าของ string 2 ข้อคือ 1. แปลงค่าเป็น string 2. แปลงค่าเป็น int. ex. แปลงค่าเป็น string จากค่าของ int.

7. คำถามทางการทดลอง

7.1. จงอธิบายความหมายของฟังก์ชันต้นแบบ (Prototype function)

คือฟังก์ชันที่บอกชื่อฟังก์ชัน และประเภทของตัวแปรที่ส่งเข้าและออกของฟังก์ชันก่อนการเรียกใช้ฟังก์ชัน เช่น main และฟังก์ชันตัวอื่น.

7.2. จงอธิบายลักษณะการจัดวางโค้ดโปรแกรมที่ดีของฟังก์ชันต้นแบบ และคำอธิบายฟังก์ชัน

คือ comment ที่บอกชื่อฟังก์ชัน และประเภทของตัวแปรที่ส่งเข้าและออกของฟังก์ชัน.

7.3. จงอธิบายว่าผู้เรียนจะเลือกสร้างและฟังก์ชันเมื่อใด เพราะเหตุใด?

เมื่อต้องการใช้ฟังก์ชันที่ซ้ำๆ กัน หรือต้องการใช้ฟังก์ชันที่ซับซ้อน.

7.4. จงอธิบายว่าหากต้องการสร้างตัวแปรเพื่อรับค่าที่ได้จากฟังก์ชัน ควรสร้างฟังก์ชันลักษณะใด เพราะเหตุใด?

ที่รับค่าคืนค่าและแสดงผล. ex. รับค่าจากฟังก์ชัน. ex. รับค่าจากฟังก์ชัน.