

ใบงานการทดลองที่ 11
เรื่อง ฟังก์ชัน

1. จุดประสงค์ทั่วไป

2.1. รู้และเข้าใจหลักภาษาที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

2.1.29. บอกและอธิบายฟังก์ชัน

2.1.30. ฝึกหัดและทดลองใช้ฟังก์ชัน

2.1.31. ออกแบบแนวทางการใช้ฟังก์ชันเพื่อให้ทำงานได้มีประสิทธิภาพมากขึ้น

2.1.32. แนะนำแนวทางการใช้ฟังก์ชันอย่างเป็นระบบ

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

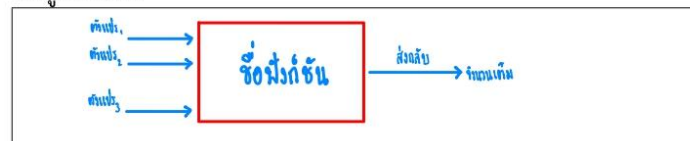
4.1. จงบอกและอธิบายความหมายของ “ฟังก์ชัน”

เป็นชุดคำสั่งที่ถูกนำมาใช้ซ้ำกันเพื่อจุดประสงค์หนึ่งใดอย่างหนึ่ง และใช้ส่งค่ากลับคืนมา

4.2. จงระบุประโยชน์ของ “ฟังก์ชัน”

ฟังก์ชัน เช่น ตัวรับ ที่สามารถนำค่าไปใช้ซ้ำได้ และลดความซ้ำซ้อน

4.3. จงยกตัวอย่างการสร้างโครงสร้างของฟังก์ชัน พร้อมวาดรูปประกอบการอธิบาย
วาดรูปประกอบ

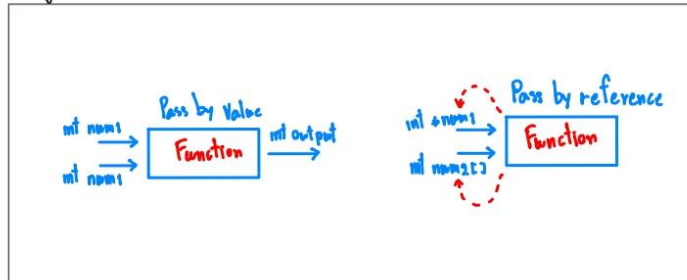


คำอธิบาย

Input ก็คือได้
Output ได้ค่าคืน

4.4. ฟังก์ชันที่มีการส่งกลับค่า และไม่มีการส่งกลับค่าแตกต่างกันอย่างไร อธิบายพร้อมวาดรูปประกอบคำอธิบาย และเขียนโค้ดตัวอย่างประกอบการอธิบาย

วาดรูปประกอบ



คำอธิบาย

ฟังก์ชันที่ไม่มีการส่งกลับ จะใช้ function void
ฟังก์ชันที่มีการส่งกลับ จะส่งกลับค่า แล้วจะวนรอบ

โค้ดโปรแกรมตัวอย่างประกอบคำอธิบาย

```
int AddXY (int x, int y) {
    return x+y ;
} //end function

Void Say Hi (int n) {
    for (int i=0 ; i<n ; i++)
        printf ("Hi\n");
} //end function
```

5. ลำดับชั้นการปฏิบัติงาน

5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

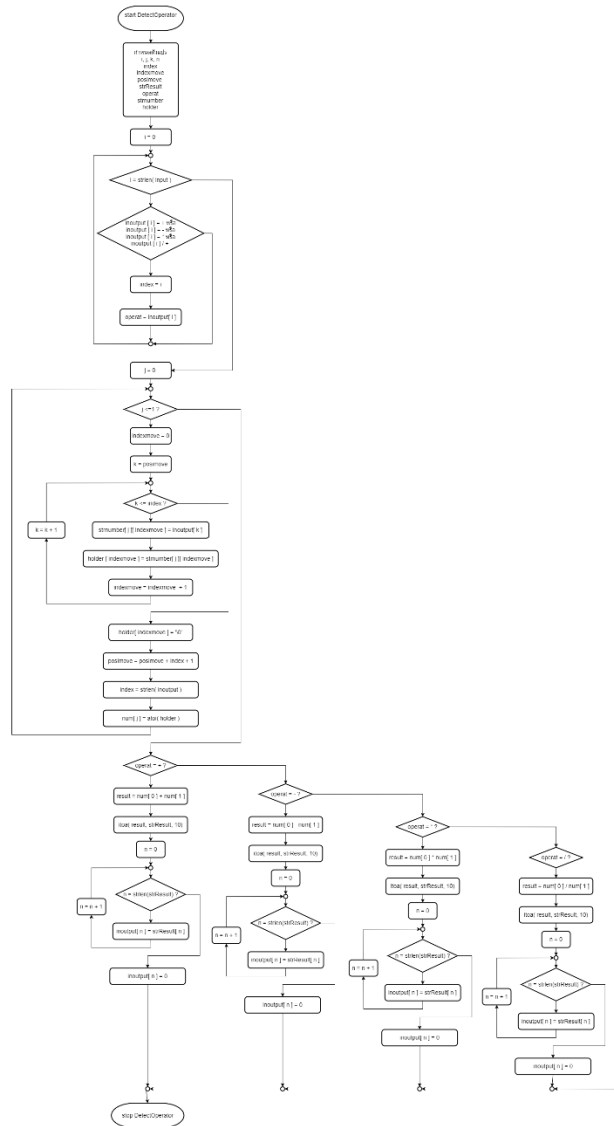
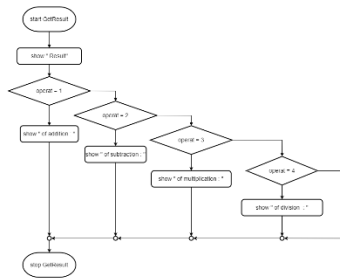
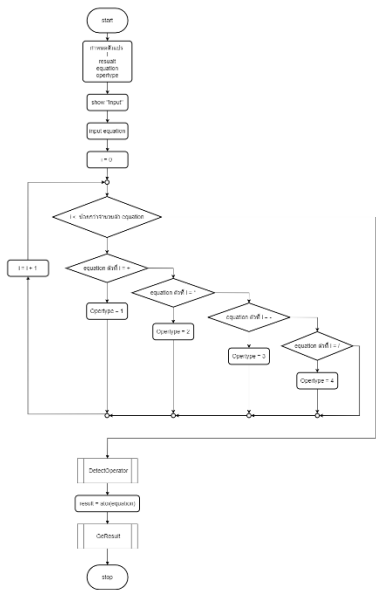
5.1.1. จงเขียนโปรแกรมเครื่องคิดเลขอย่างง่าย ด้วยการรับชุดข้อความจากผู้ใช่ เพื่อนำมาหาผลลัพธ์ของคำตอบ โดยกำหนดให้ภายในฟังก์ชันหลักจะต้องมีการเรียกใช้ฟังก์ชันการทำงานย่อยเพื่อรับค่าจากผู้ใช่ ประมวลผลตัวดำเนินการ และแสดงผลลัพธ์ ดังแสดงฟังก์ชันต้นแบบดังนี้

```
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




```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
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         else if ( equation[i] == '*' ){
23             opertype = 3 ;
24         }
25         else if ( equation[i] == '/' ){
26             opertype = 4 ;
27         }
28     }
29     DetectOperator( equation ) ;
30     result = atoi( equation ) ;
31     GetResult( result, opertype ) ;
32     return 0 ;
33 }
34
35 char DetectOperator( char input [ ] ) {
36     int i, j, k, n ;
37     int index, indexmove ;
38     int posimove = 0 ;
39     char strResult[ 100 ] ;
40     char operat, strnumber[ 10 ][ 100 ] , holder[ 100 ] ;
41     int num[ 100 ], result ;
42     for( i = 0 ; i < strlen( input ) ; i++ ) {
43         if( input[ i ] == '+' || input[ i ] == '-' || input[ i ] == '*' || input[ i ] == '/' ) {
44             index = i ;
45             operat = input[ 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

```

```

95  int GetResult( int number, int operat ) {
96      printf( "Result" ) ;
97      if ( operat == 1 ) {
98          printf( " of addition : %d ", number ) ;
99      }
100     else if ( operat == 2 ) {
101         printf( " of subtraction : %d ", number ) ;
102     }
103     else if ( operat == 3 ) {
104         printf( " of multiplication : %d ", number ) ;
105     }
106     else if ( operat == 4 ) {
107         printf( " of division : %d ", number ) ;
108     }
109     return 0 ;
110 }

```

6. สรุปผลการปฏิบัติงาน

จากบททดลองเราได้เห็นว่า ฟังก์ชัน คืออะไร และ ฟังก์ชันคืออะไร

7. คำถามทางการทดลอง

7.1. จงอธิบายความหมายของฟังก์ชันต้นแบบ (Prototype function)

เมื่อให้โปรแกรมเมอร์เขียนฟังก์ชันในโปรแกรม

7.2. จงอธิบายลักษณะการจัดวางโค้ดโปรแกรมที่ดีของฟังก์ชันต้นแบบ และคำอธิบายฟังก์ชัน

เขียนฟังก์ชัน (Prototype) ก่อน แล้วจึงเขียนฟังก์ชัน (Function) เมื่อใช้โปรแกรมเมอร์เขียนฟังก์ชัน (Prototype)

7.3. จงอธิบายว่าผู้เรียนจะเลือกสร้างและฟังก์ชันเมื่อใด เพราะเหตุใด?

เมื่อมีอะไรที่ต้องใช้ > 9 ครั้ง เพราะ ทำได้โดยไม่ต้องเขียนซ้ำ

7.4. จงอธิบายว่าหากต้องการสร้างตัวแปรเพื่อรับค่าที่ได้จากฟังก์ชัน ควรสร้างฟังก์ชันลักษณะใด เพราะเหตุใด ?

ควรสร้าง function ที่ให้ void เพราะ void ไม่สามารถส่งค่าออกไปได้ ไม่ใช่ array หรือ pointer