

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

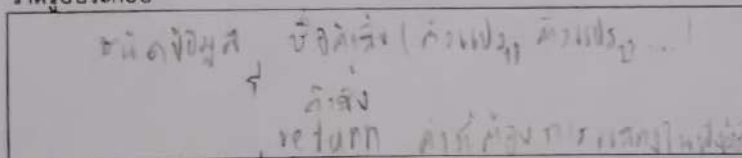
4.1. จงบอกและอธิบายความหมายของ "ฟังก์ชัน"

ฟังก์ชัน คือชุดคำสั่งที่สามารถเรียกใช้งานซ้ำได้
หรือที่รู้จักกันในชื่อ ฟังก์ชัน และ output

4.2. จงระบุประโยชน์ของ "ฟังก์ชัน"

ในการใช้คำสั่งที่ซ้ำกันหลายครั้ง สามารถใช้งาน
คำสั่งที่ปรกติได้ เช่น การคำนวณ

4.3. จงยกตัวอย่างการสร้างโครงสร้างของฟังก์ชัน พร้อมวาดรูปประกอบการอธิบาย
วาดรูปประกอบ



คำอธิบาย

ชนิดข้อมูล คือ ชนิดค่าที่ได้จากการ return หรือที่

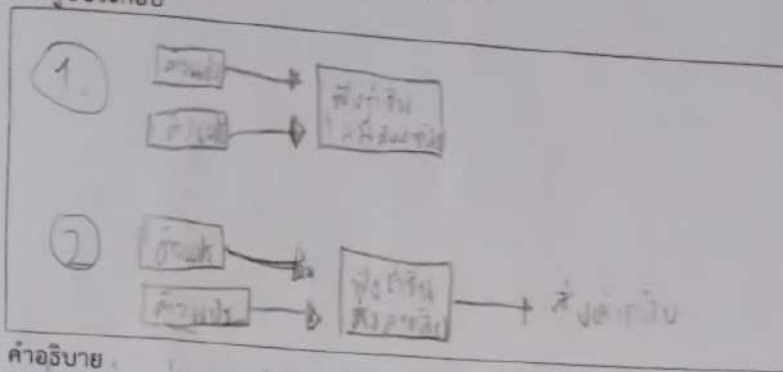
เรียกว่า return ค่าตามค่าในวงเล็บที่ return

ชื่อฟังก์ชัน คือ ชื่อที่ใช้เรียกใช้งาน Function ค่าแปรคือ parameter ที่ส่งเข้าไปในฟังก์ชัน

return คือค่าที่ส่งกลับจากฟังก์ชัน

4.4 ฟังก์ชันที่มีการส่งกลับค่า และไม่มีการส่งกลับค่าแตกต่างกันอย่างไร อธิบายพร้อมวาดรูปประกอบคำอธิบาย และเขียนโค้ดตัวอย่างประกอบการอธิบาย

วาดรูปประกอบ



คำอธิบาย

ส่วนที่ 1 คือ ฟังก์ชันที่ไม่มีการส่งกลับค่า แสดงว่า 1. เมื่อถูกใช้งานแล้วจะไม่ส่งกลับค่าให้ฟังก์ชัน ค่าที่ส่งกลับจะอยู่ในตัวแปรที่เราได้ส่งค่าไปให้ฟังก์ชัน

ส่วนที่ 2 คือ ฟังก์ชันที่ส่งกลับค่า เป็นการที่ฟังก์ชันส่งกลับค่าที่คำนวณได้ผ่านฟังก์ชัน return โดยที่เราจะส่งค่าที่คำนวณได้กลับมาให้

โค้ดโปรแกรมตัวอย่างประกอบคำอธิบาย

```

① ฟังก์ชันไม่มีการส่งกลับ
void Doit(Para1, Para2) {
    // โค้ด
}

② ฟังก์ชันส่งกลับ
int Doit(Para1, Para2) {
    int a;
    // โค้ด
    return a;
}

void Doit(Para1, Para2) {
    // โค้ด
    Para1 = a; // ไม่ส่งกลับค่า
}

```

5. ลำดับขั้นการปฏิบัติงาน

5.1 จดเขียนผังงานและได้โปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

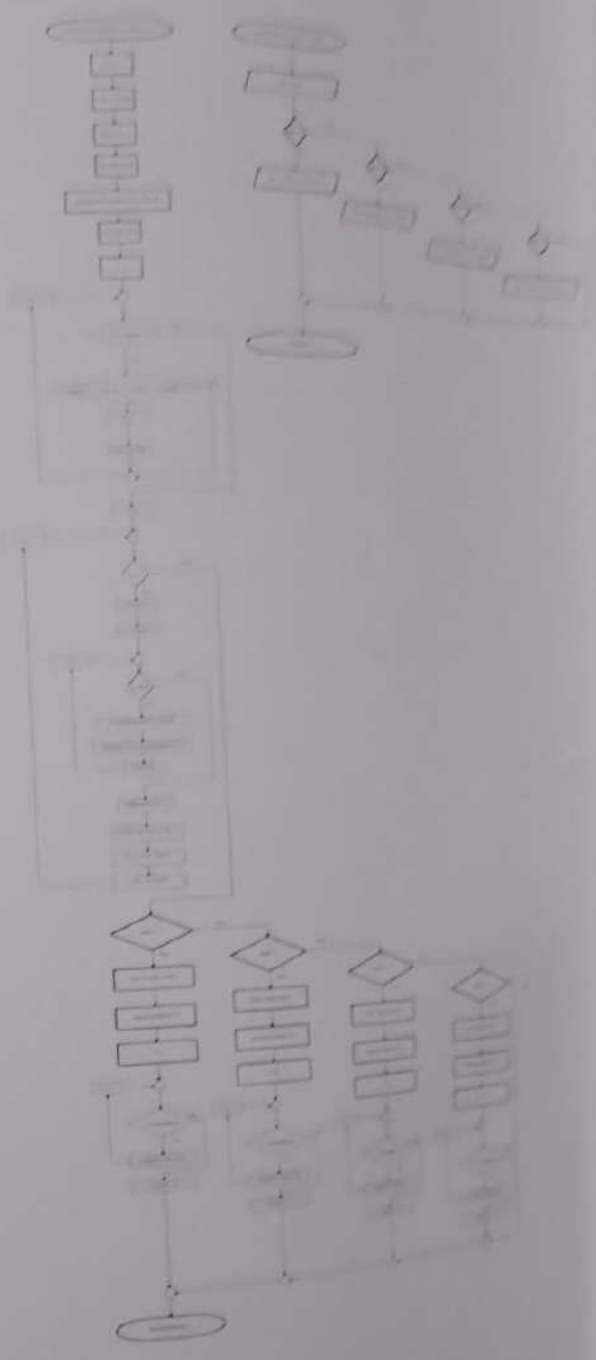
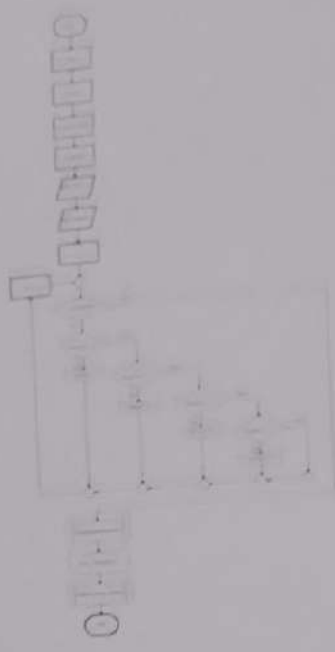
5.1.1 จดเขียนโปรแกรมเครื่องคิดเลขอย่างง่าย ด้วยการรับข้อมูลความจากผู้ใช้ เพื่อคำนวณหาผลลัพธ์ของค่าคูณ โดยกำหนดให้ภายในฟังก์ชันหลักจะต้องมีการเรียกใช้ฟังก์ชันการทำงานย่อยเพื่อรับค่าจากผู้ใช้ ประมวลผลตัวดำเนินการ และแสดงผลลัพธ์ ดังแสดงฟังก์ชันต้นแบบดังนี้

```
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 จดเขียนผังงาน

ผังงาน

5.1.3 จดเขียนโค้ดโปรแกรม

โค้ดโปรแกรม

Lab11

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

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

```
int main() {
    int i;
    int result;
    char equation[100];
    int opertype;
    printf( "Input : " );
    gets( equation );
    for( i = 0 ; i < strlen( equation ) ; i++ ) {
        if( equation[i] == '+' ){
            opertype = 1 ;
        }
        else if ( equation[i] == '-' ){
            opertype = 2 ;
        }
        else if ( equation[i] == '*' ){
            opertype = 3 ;
        }
        else if ( equation[i] == '/' ){
            opertype = 4 ;
        }
    }
    DetectOperator(equation);
    result = atoi(equation);
}
```

```
GetResult(result, opertype);  
return 0;  
}
```

```
char DetectOperator( char inoutput [] ) {  
    int i, j, k, n;
```

```
    int index, indexmove;
```

```
    int posimove = 0;
```

```
    char strResult[100];
```

```
    char operat, strnumber[10][100], holder[100];
```

```
    int num[100], result;
```

```
    for( i = 0; i < strlen( inoutput ); i++ ) {
```

```
        if( inoutput[i] == '+' || inoutput[i] == '-' || inoutput[i] == '*' || inoutput[i] == '/' ) {
```

```
            index = i;
```

```
            operat = inoutput[i];
```

```
        }
```

```
    }
```

```
    for( j = 0; j <= 1; j++ ) {
```

```
        indexmove = 0;
```

```
        for( k = posimove; k < index; k++ ) {
```

```
            strnumber[j][indexmove] = inoutput[k];
```

```
            holder[indexmove] = strnumber[j][indexmove];
```

```
            indexmove++;
```

```
        }
```

```
        holder[indexmove] = "\0";
```

```
        posimove = posimove + index + 1;
```

```
        index = strlen( inoutput );
```

```
        num[j] = atoi(holder);
```

```
    }
```

```
    if ( operat == '+' ) {
```

```

        result = num[0] + num[1];
        itoa(result, strResult, 10);
        for ( n = 0 ; n < strlen(strResult) ; n++ ) {
            inoutput[n] = strResult[n];
        }
        inoutput[n] = '\0';
    }

    else if ( operat == '-' ){
        result = num[0] - num[1];
        itoa(result, strResult, 10);
        for ( n = 0 ; n < strlen(strResult) ; n++ ) {
            inoutput[n] = strResult[n];
        }
        inoutput[n] = '\0';
    }

    else if ( operat == '*' ){
        result = num[0] * num[1];
        itoa(result, strResult, 10);
        for ( n = 0 ; n < strlen(strResult) ; n++ ) {
            inoutput[n] = strResult[n];
        }
        inoutput[n] = '\0';
    }

    else if ( operat == '/' ){
        result = num[0] / num[1];
        itoa(result, strResult, 10);
        for ( n = 0 ; n < strlen(strResult) ; n++ ) {
            inoutput[n] = strResult[n];
        }
        inoutput[n] = '\0';
    }
}

```



```
}  
int GetResult( int number, int operat ){  
    printf( "Result" );  
    if (operat == 1) {  
        printf(" of addition : %d ", number);  
    }  
    else if (operat == 2) {  
        printf(" of subtraction : %d ", number);  
    }  
    else if (operat == 3) {  
        printf(" of multiplication : %d ", number);  
    }  
    else if (operat == 4) {  
        printf(" of division : %d ", number);  
    }  
    return 0;  
}
```

6. สรุปผลการปฏิบัติงาน

จากการปฏิบัติจะพบว่ามีการใช้ฟังก์ชัน atoi ที่แปลงค่า string เป็น integer และมีการใช้ฟังก์ชัน itoa ที่แปลงค่า int เป็น string เพื่อใช้ในการส่งค่าส่งกลับ string ในส่วนที่คำนวณและค่า และการส่งค่าส่งกลับอีก 2 ฟังก์ชัน เพื่อตรวจสอบเงื่อนไขการคำนวณและค่าออกมา แล้วส่งค่าส่งกลับค่าที่อยู่ที่ชี้ไปยังส่วนที่คำนวณ

7. คำถามทางการทดลอง

7.1 จงอธิบายความหมายของฟังก์ชันต้นแบบ (Prototype function)

การประกาศ ฟังก์ชันก่อนที่ระบบการทำงานจะอยู่ในของฟังก์ชัน เพื่อที่นักพัฒนาฟังก์ชันที่จะใช้ระบบการทำงานนั้นส่งค่าส่งกลับค่า

7.2 จงอธิบายลักษณะการจัดวางโค้ดโปรแกรมที่ดีของฟังก์ชันต้นแบบ และคำอธิบายฟังก์ชัน

การใช้ comment อธิบายหน้าที่ของฟังก์ชันนั้นๆ ได้เข้าใจได้อย่างละเอียด

7.3 จงอธิบายว่าผู้เรียนจะเลือกสร้างและฟังก์ชันเมื่อใด เพราะเหตุใด?

เมื่อต้องการให้โค้ดสั้นๆ ได้ใจความ และดูง่ายต่อการอ่าน และใช้โค้ดซ้ำๆ กัน

7.4 จงอธิบายว่าหากต้องการสร้างตัวแปรเพื่อรับค่าที่ได้จากฟังก์ชัน ควรสร้างฟังก์ชันลักษณะใด

เพราะเหตุใด?

ฟังก์ชันนั้นต้องกลับค่าออกมาได้ ซึ่งจากตัวอย่าง Assignment ของฟังก์ชันนั้นกับตัวแปรนั้นได้ทันที ซึ่งจากที่คำสั่งของฟังก์ชัน