

ใบงานการทดลองที่ 11  
เรื่อง ฟังก์ชัน

1. จุดประสงค์ทั่วไป

2.1. รู้และเข้าใจหลักภาษาที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

2.1.29. บอกและอธิบายฟังก์ชัน

2.1.30. ลึกหัดและทดลองใช้ฟังก์ชัน

2.1.31. ออกแบบแนวทางการใช้ฟังก์ชันเพื่อให้ทำงานมีประสิทธิภาพมากขึ้น

2.1.32. แนะนำแนวทางการใช้ฟังก์ชันอย่างเป็นระบบ

3. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

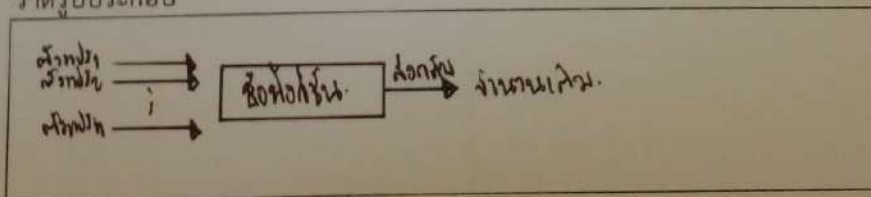
4.1. จงบอกและอธิบายความหมายของ "ฟังก์ชัน"

เป็นการรวบรวมคำสั่งที่ซ้ำกัน เพื่อลดความซับซ้อนให้กับโปรแกรม  
ทำงานซ้ำ

4.2. จงระบุประโยชน์ของ "ฟังก์ชัน"

ลดความซับซ้อนของโปรแกรมได้ไม่จำกัด

4.3. จงยกตัวอย่างการสร้างโครงสร้างของฟังก์ชัน พร้อมวาดรูปประกอบการอธิบาย  
วาดรูปประกอบ

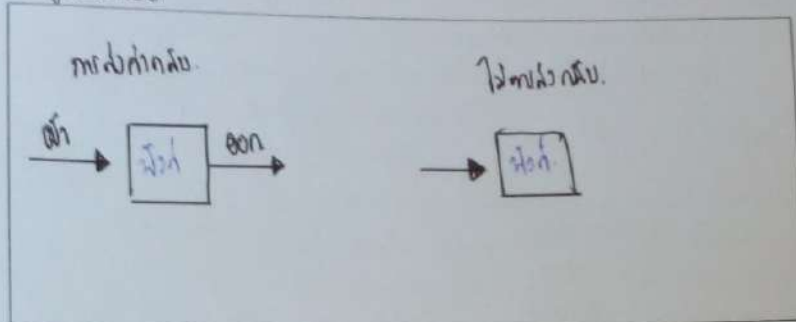


คำอธิบาย

กำหนดตัวแปร ชื่อฟังก์ชัน ส่งกลับ เมื่อได้จำนวนเต็มออก

4.4. ฟังก์ชันที่มีการส่งกลับค่า และไม่มีการส่งกลับค่าแตกต่างกันอย่างไร อธิบายพร้อมวาดรูปประกอบคำอธิบาย และเขียนโค้ดตัวอย่างประกอบการอธิบาย

วาดรูปประกอบ



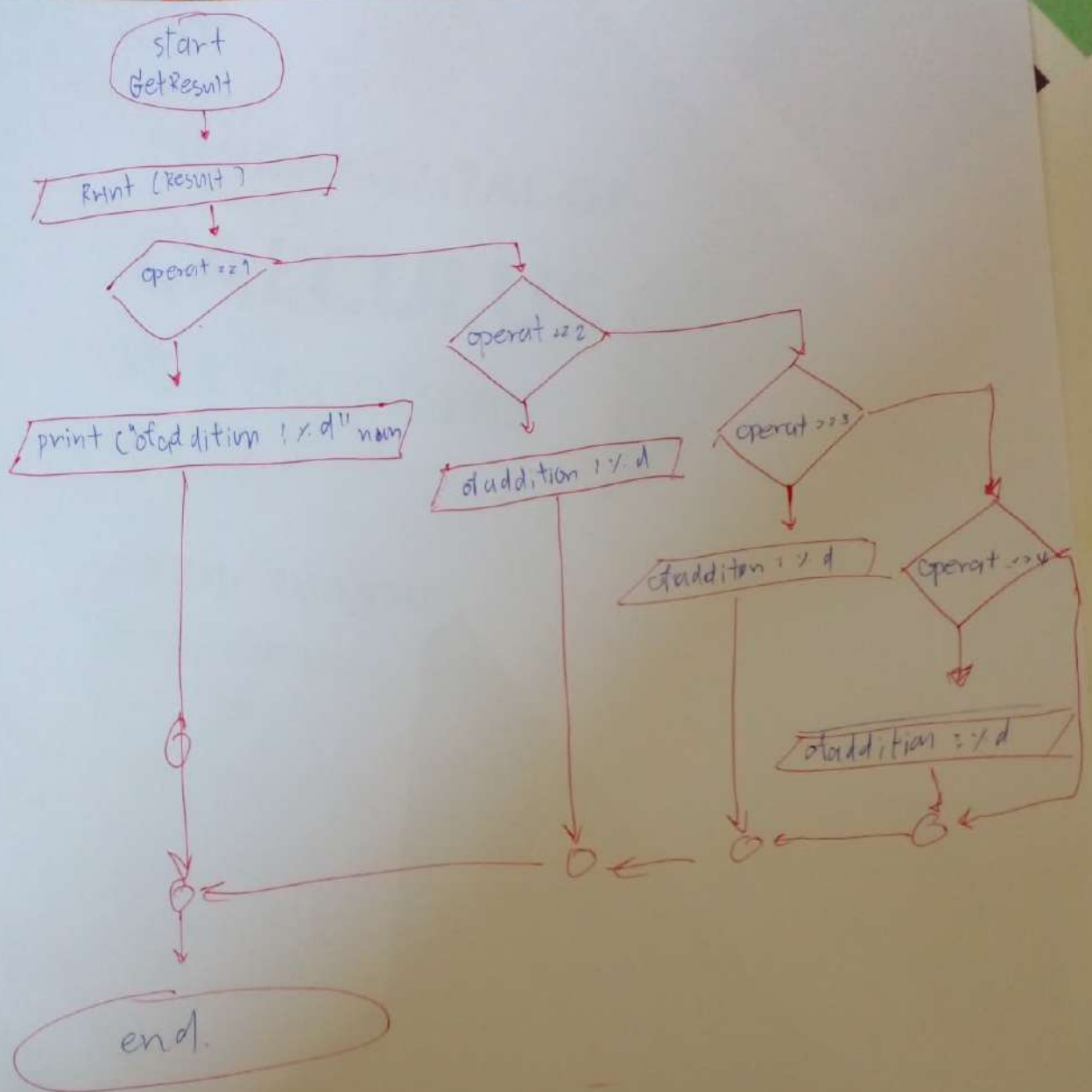
คำอธิบาย

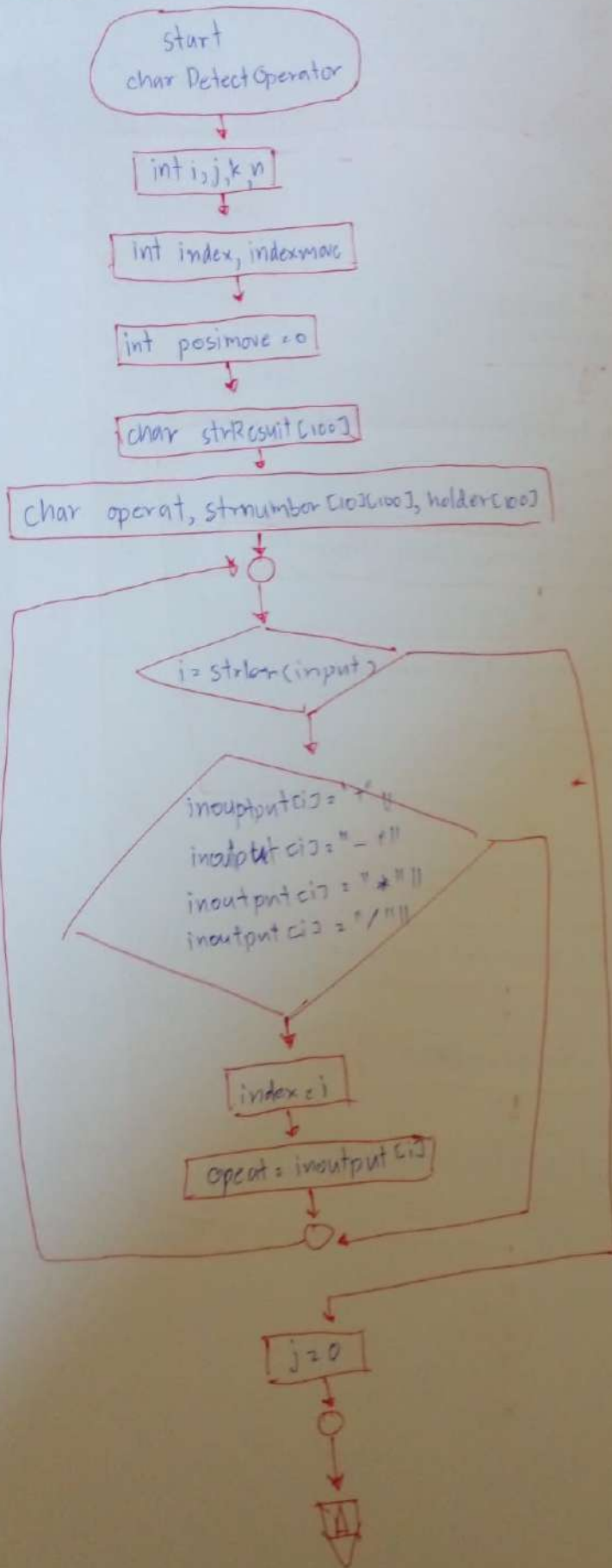
ฟังก์ชันส่งกลับค่า เป็นฟังก์ชันที่ส่งค่ากลับเพื่อทำงานได้จริงเป็นฟังก์ชันที่ส่งค่าออกมาให้สามารถไปใช้ต่อฟังก์ชันอื่นๆได้เช่นฟังก์ชันหาผลบวกของตัวเลข

ฟังก์ชันไม่ส่งกลับค่า เป็นฟังก์ชันที่ทำงานแล้วไม่ส่งค่ากลับออกมาให้ทำงานต่อได้เช่นฟังก์ชันหาผลบวกของตัวเลขโดยไม่ได้ส่งค่ากลับออกมาให้ทำงานต่อได้

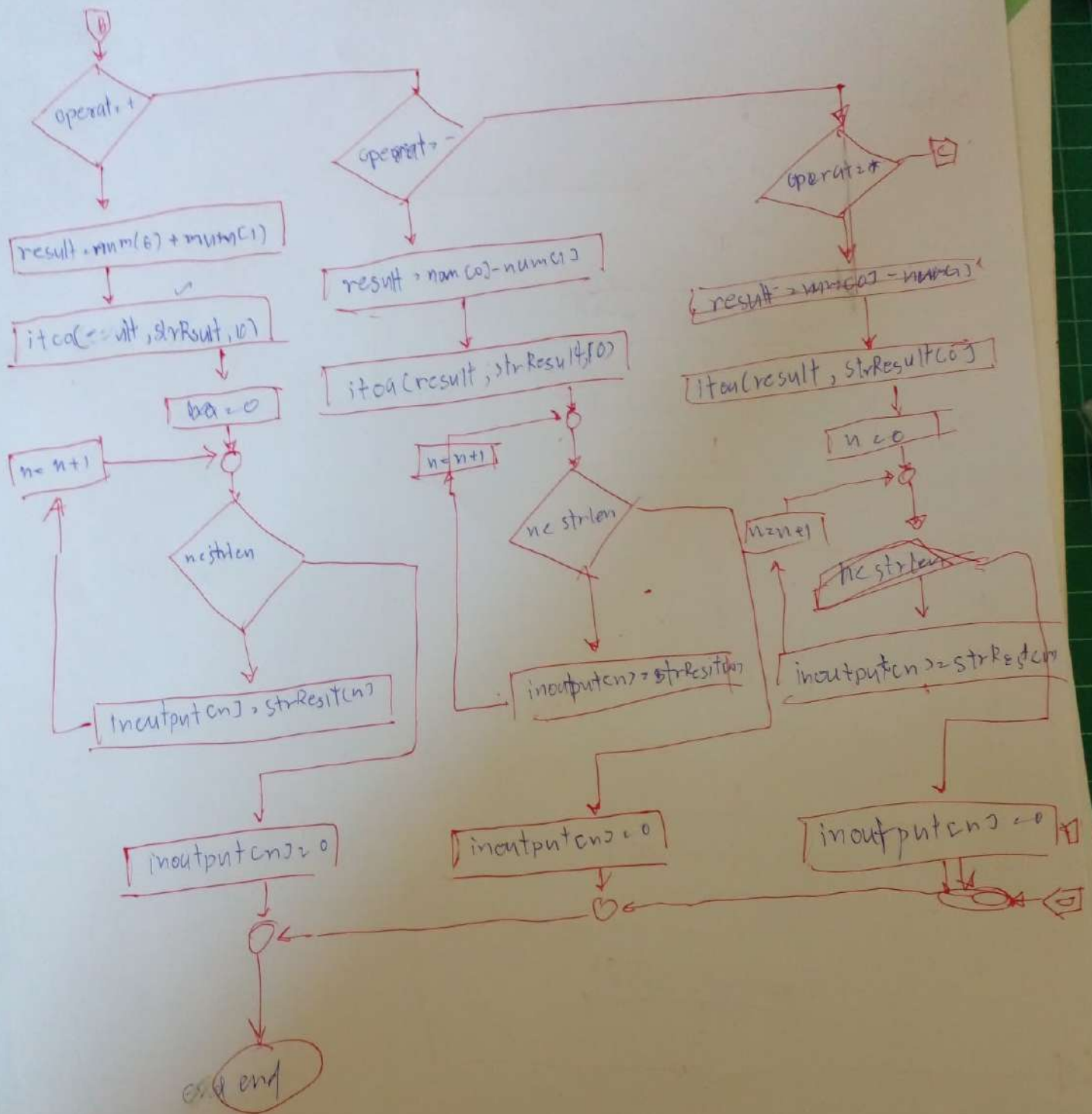
โค้ดโปรแกรมตัวอย่างประกอบคำอธิบาย

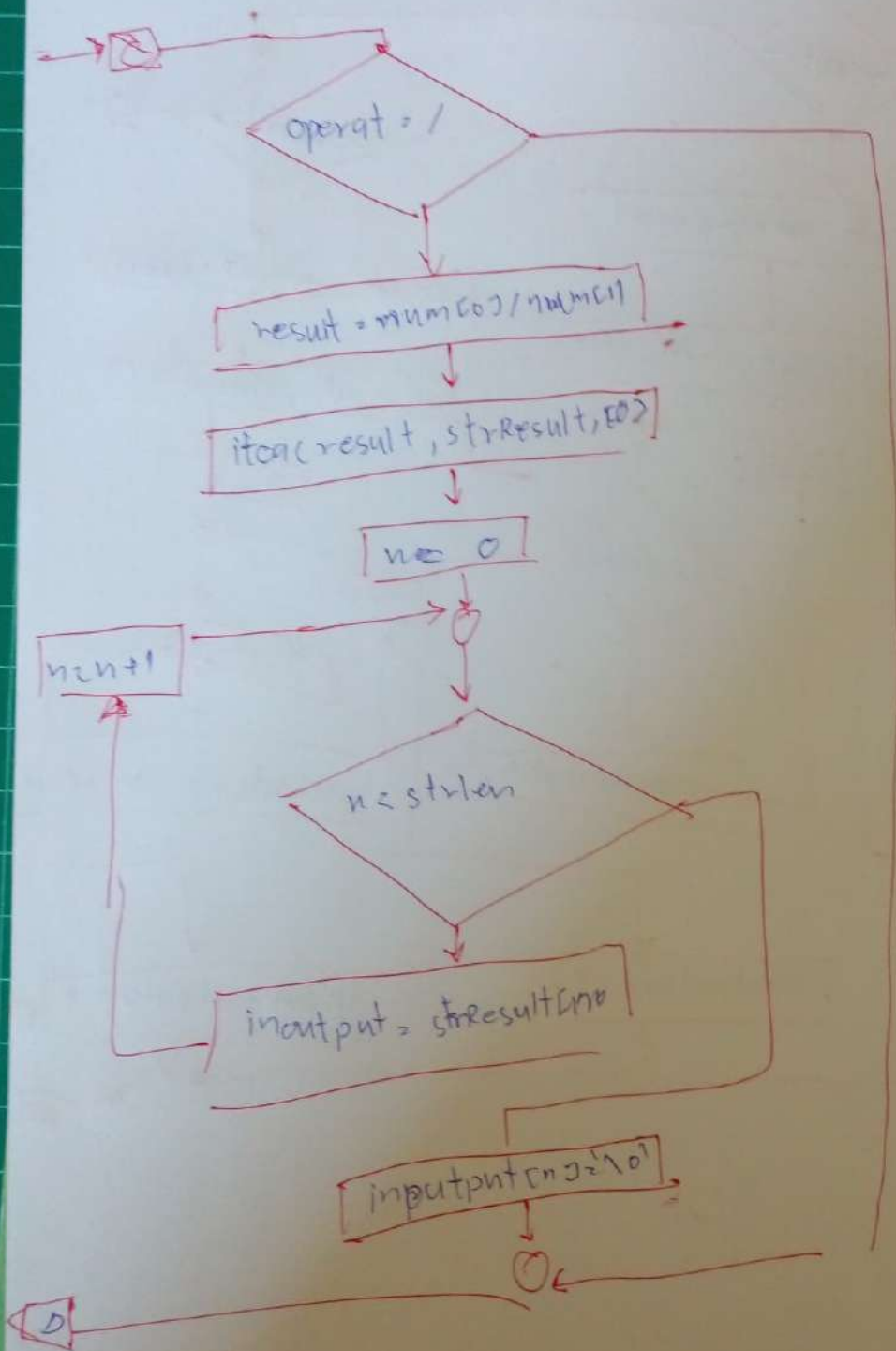
<p>ฟังก์ชัน 1</p> <p>ตัวแปรภายในฟังก์ชัน</p> <p>ค่าส่ง</p> <p>return</p> <p>↓</p>	<p>ฟังก์ชัน 2</p> <p>ตัวแปรภายในฟังก์ชัน</p> <p>ค่าส่ง</p> <p>↓</p>
---	---

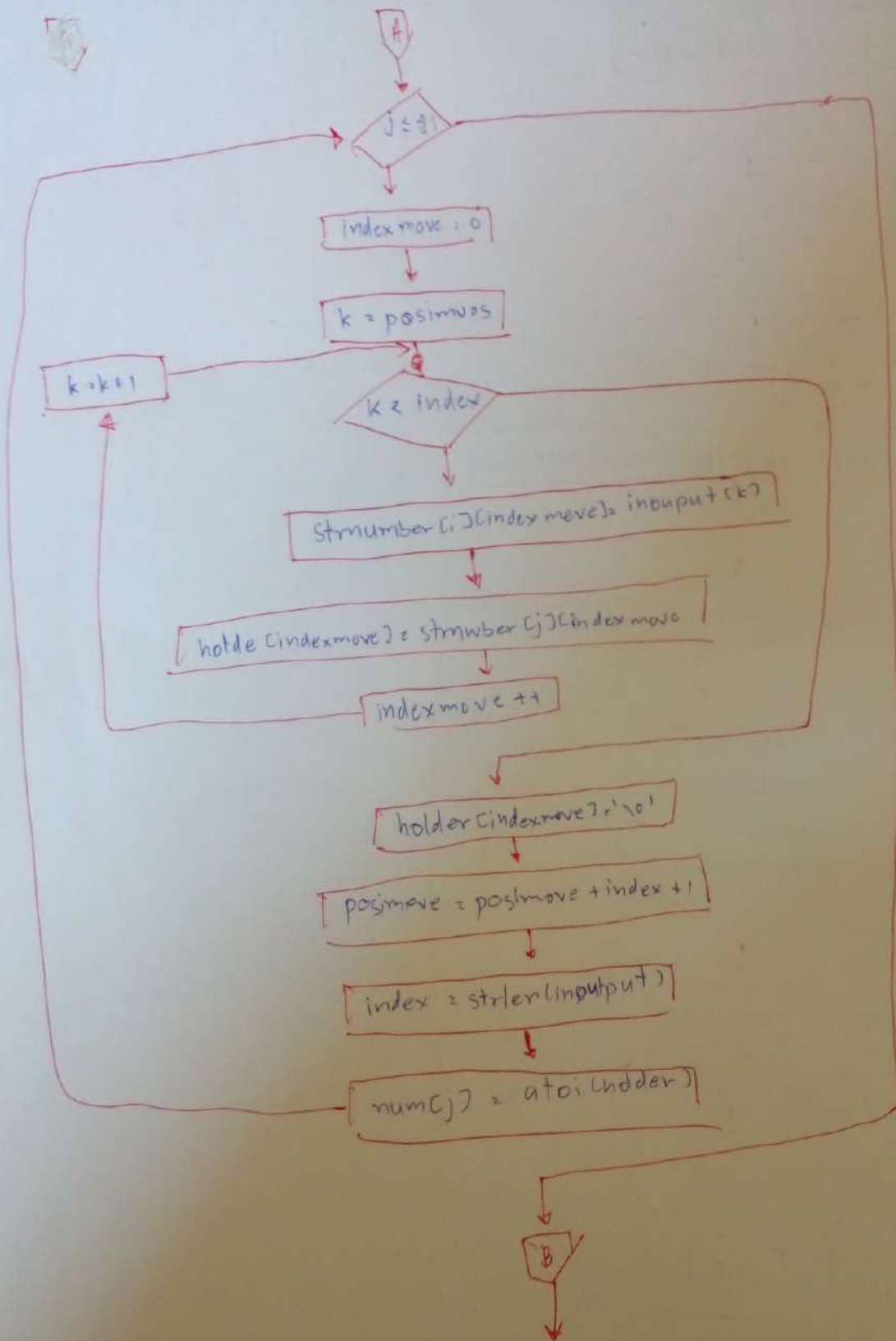






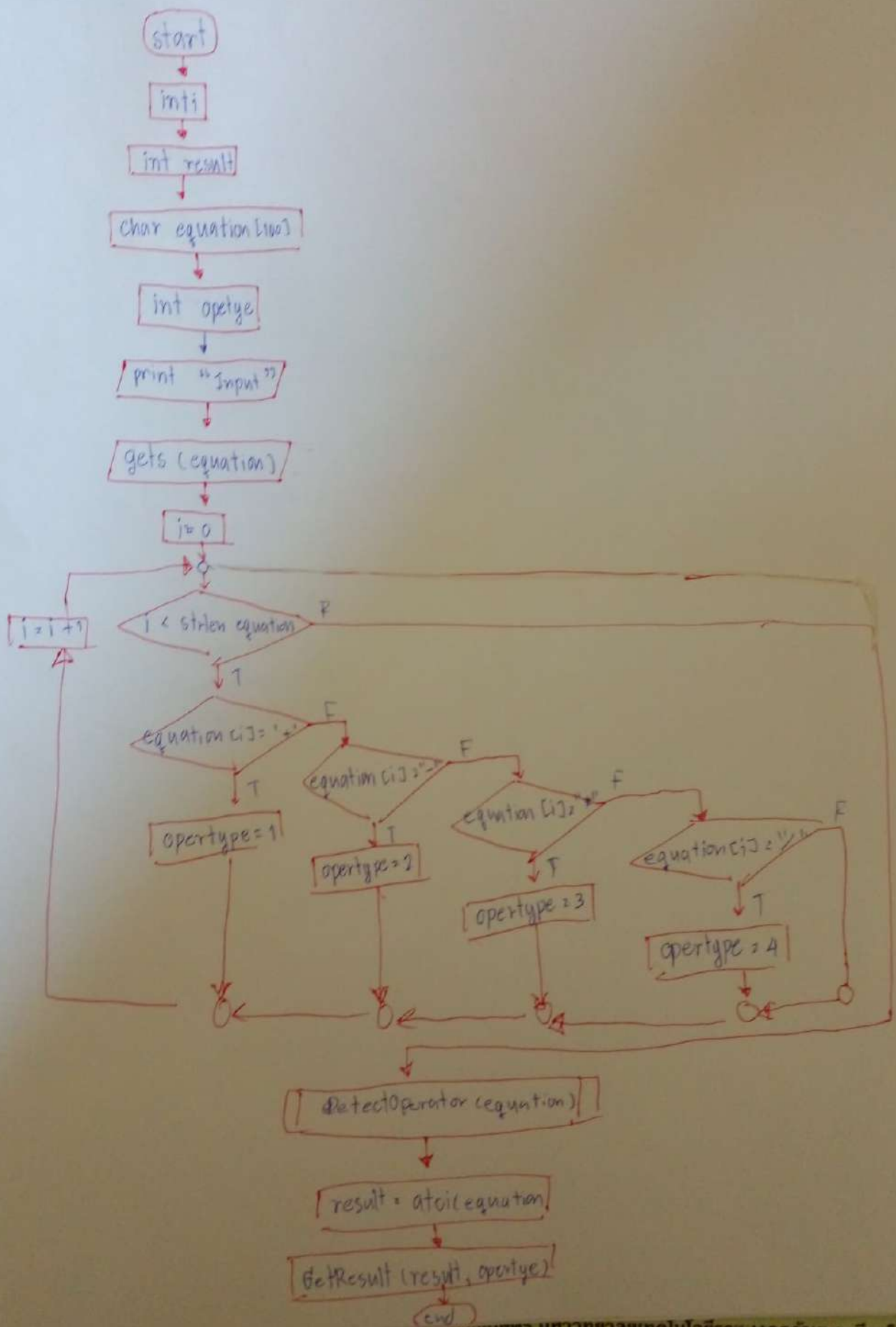








# ผังงานที่ 11





```

    }

    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 ( h = 0 ; h < strlen(strResult) ; h++ ) {

        inoutput[h] = strResult[h];

    }

    inoutput[h] = '\0' ;

}

else if ( operat == '*' ){

    result = num[0] * num[1] ;

    itoa(result, strResult, 10) ;

    for ( h = 0 ; h < strlen(strResult) ; h++ ) {

        inoutput[h] = strResult[h];

    }

    inoutput[h] = '\0' ;

}

else if ( operat == '/' ){

    result = num[0] / num[1] ;

    itoa(result, strResult, 10) ;

    for ( h = 0 ; h < strlen(strResult) ; h++ ) {

        inoutput[h] = strResult[h];

    }

    inoutput[h] = '\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 ;
```

```
}
```

```
#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 ;
```

## 6. สรุปผลการปฏิบัติงาน

### 7. คำถามทางการทดลอง

7.1. จงอธิบายความหมายของฟังก์ชันต้นแบบ (Prototype function)  
เป็นประกาศก่อนคอมไพล์ว่าฟังก์ชันอะไร จะทำงานอย่างไร

7.2. จงอธิบายลักษณะการจัดวางโค้ดโปรแกรมที่ดีของฟังก์ชันต้นแบบ และคำอธิบายฟังก์ชัน  
ควรใช้ `function main` แทนฟังก์ชัน `error`.

7.3. จงอธิบายว่าผู้เรียนจะเลือกสร้างและฟังก์ชันเมื่อใด เพราะเหตุใด?  
เมื่อ `function` ถูกใช้ภายใน `main` จะทำให้ผู้จัดทำโค้ดเกิดความสับสน  
และอาจก่อให้เกิด `error` ได้

7.4. จงอธิบายว่าหากต้องการสร้างตัวแปรเพื่อรับค่าที่ได้จากฟังก์ชัน ควรสร้างฟังก์ชันลักษณะใด  
เพราะเหตุใด?

ควรสร้างในฟังก์ชันที่กำหนดให้ใช้ `function` ชื่อฟังก์ชัน `function`  
เพราะสามารถรับค่าที่ส่งกลับ