

ใบงานการทดลองที่ 6

เรื่อง การวนรอบโดยใช้คำสั่ง for

1. จุดประสงค์ทั่วไป

- 2.1. รู้และเข้าใจหลักภาษาที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์

2. จุดประสงค์เชิงพฤติกรรม

- 2.1.17. บอกและอธิบายคำสั่งวนรอบ
- 2.1.18. ฝึกหัดและทดลองใช้คำสั่งวนรอบ
- 2.1.19. ออกแบบแนวทางการใช้คำสั่งวนรอบเพื่อให้ทำงานได้มีประสิทธิภาพมากขึ้น
- 2.1.20. แนะนำแนวทางการใช้คำสั่งวนรอบอย่างเป็นระบบ

3. เครื่องมือและอุปกรณ์

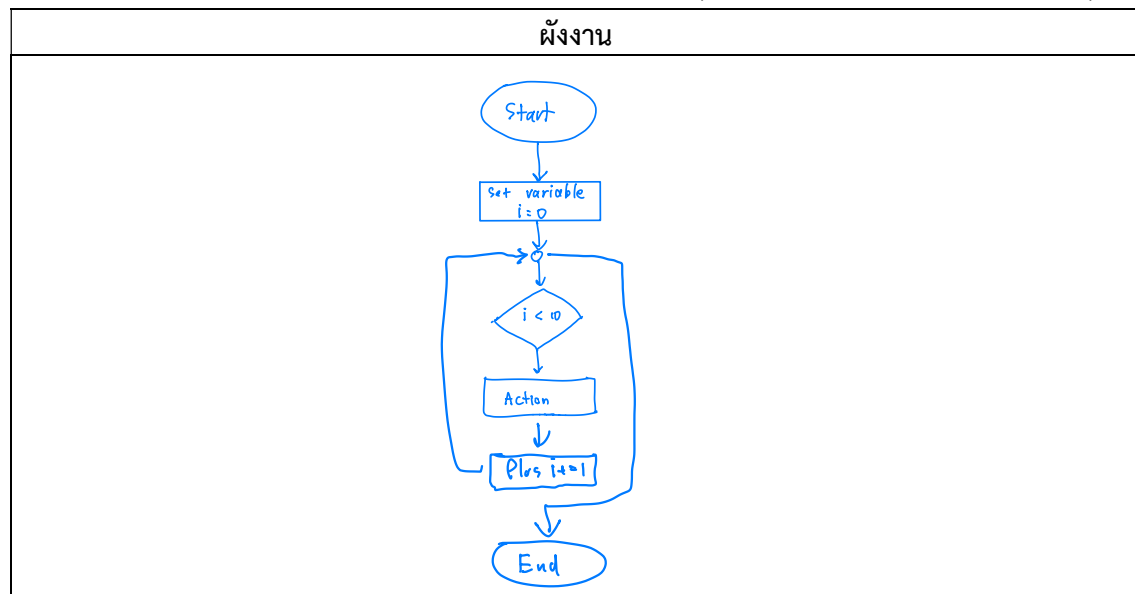
เครื่องคอมพิวเตอร์ 1 เครื่องที่ติดตั้งโปรแกรม Dev-C

4. ทฤษฎีการทดลอง

- 4.1. จงบอกและอธิบายการทำงานของคำสั่ง for (ให้เห็นองค์ประกอบครบทั้ง 3 สถานะ)

for loop แสดงในรูปแนวทางแสดงใน parameter จะประกอบด้วย 3 องค์ประกอบ คือ Initial state ; Condition state ; และ Increment state โดยส่วนแรกจะกำหนดตัวแปรที่จะวนซ้ำ ส่วนที่สองจะกำหนดเงื่อนไขที่จะวนซ้ำ และส่วนที่สามจะกำหนดว่าจะเพิ่มค่าตัวแปรในครั้งต่อไปหรือไม่

- 4.2. จงวาดผังงานแสดงกระบวนการทำงานของคำสั่ง for (ให้เห็นองค์ประกอบครบทั้ง 3 สถานะ)



```

// Accessing 2D Array
int arr[5] = {1, 2, 3, 4, 5};           Output:
for (int loop = 0; loop < 5; loop++) {  1 2 3 4 5
    printf("%d", array[loop]);
}

```

5.1. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.1.1. จงเขียนโปรแกรมเพื่อรับจำนวนบรรทัดจากผู้ใช้ และแสดงผลลัพธ์เป็นรูปต่อไปนี้

Test Case		Test Case	
Input	Input your line : 5	Input	Input your line : 7
Output	<pre> * * * * * * * * * * * * * * *</pre>	Output	<pre> *</pre>

ผังงาน

```
graph TD; Start([Start]) --> GetInput[/Get input from Keyboard "max_line"/]; GetInput --> SetLine[Set variable line = 1]; SetLine --> Decision1{line <= max_line}; Decision1 -- True --> SetI[Set variable i = 0]; Decision1 -- False --> Decision2{i < line}; SetI --> Join(( )); Decision2 -- True --> OutputStar[/Output "*" without new line/]; OutputStar --> PlusI[Plus "i" variable to 1]; PlusI --> Decision2; Decision2 -- False --> PlusLine[Plus "line" variable to 1]; PlusLine --> OutputBlank[/Output blank with new line/]; OutputBlank --> Decision1; Join --> End([End]);
```

5.1.3. โค้ดโปรแกรม

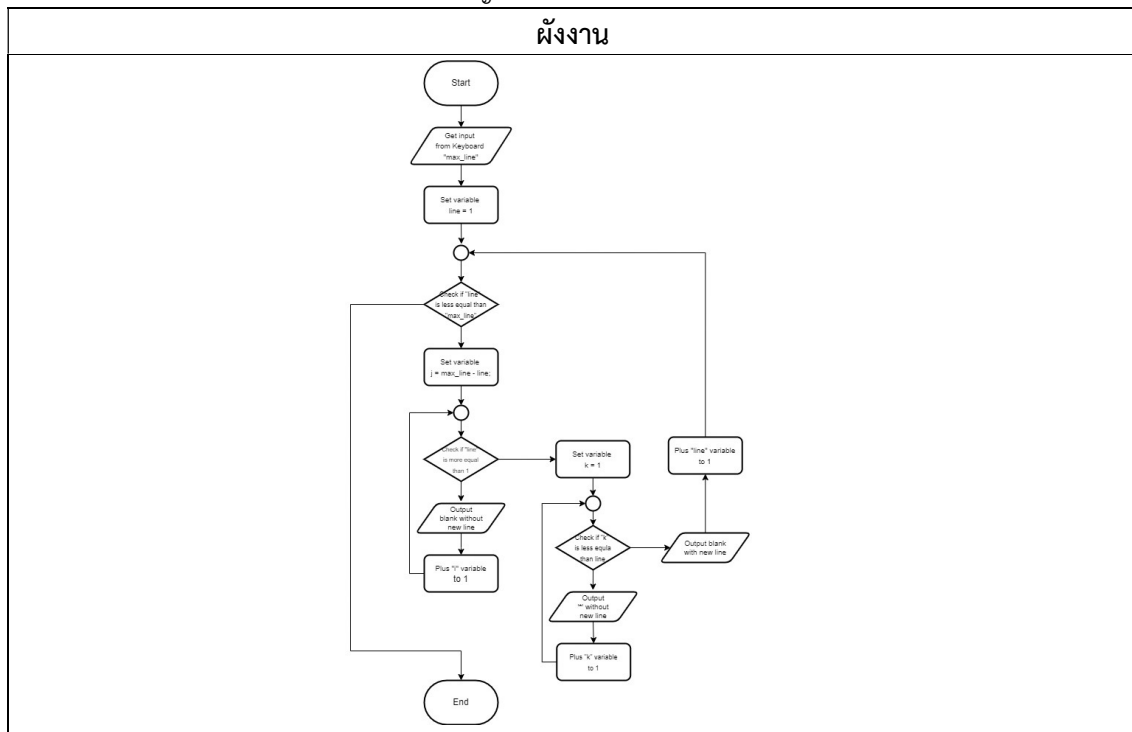
โค้ดโปรแกรม
<pre>#include <stdio.h> int main() { int i = 0 ; int j = 0 ; int l = 0 ; printf (" Enter Lines = "); scanf ("%d", &l); for (i = 1 ; i<= l ; i++){ for (j=1 ; j<=i ; j++){ printf(" * "); } //end for printf(" \n"); } //end for return 0 ; } // end function</pre>

5.2. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.2.1. จงเขียนโปรแกรมเพื่อรับจำนวนบรรทัดจากผู้ใช้ และแสดงผลลัพธ์เป็นรูปต่อไปนี้

Test Case		Test Case	
Input	Input your line : 5	Input	Input your line : 7
Output	<pre>***** * * * * * * * * * * </pre>	Output	<pre>***** * * * * * * * * * * * * </pre>

5.2.2. ผังงานเพื่อแก้โจทย์ปัญหา



5.2.3. โค้ดโปรแกรม

โค้ดโปรแกรม

```

#include <stdio.h>

int main(){
    int i = 0;
    int j = 0;
    int l = 0;
    printf ( " Enter Lines = ");
    scanf ("%d", &l);

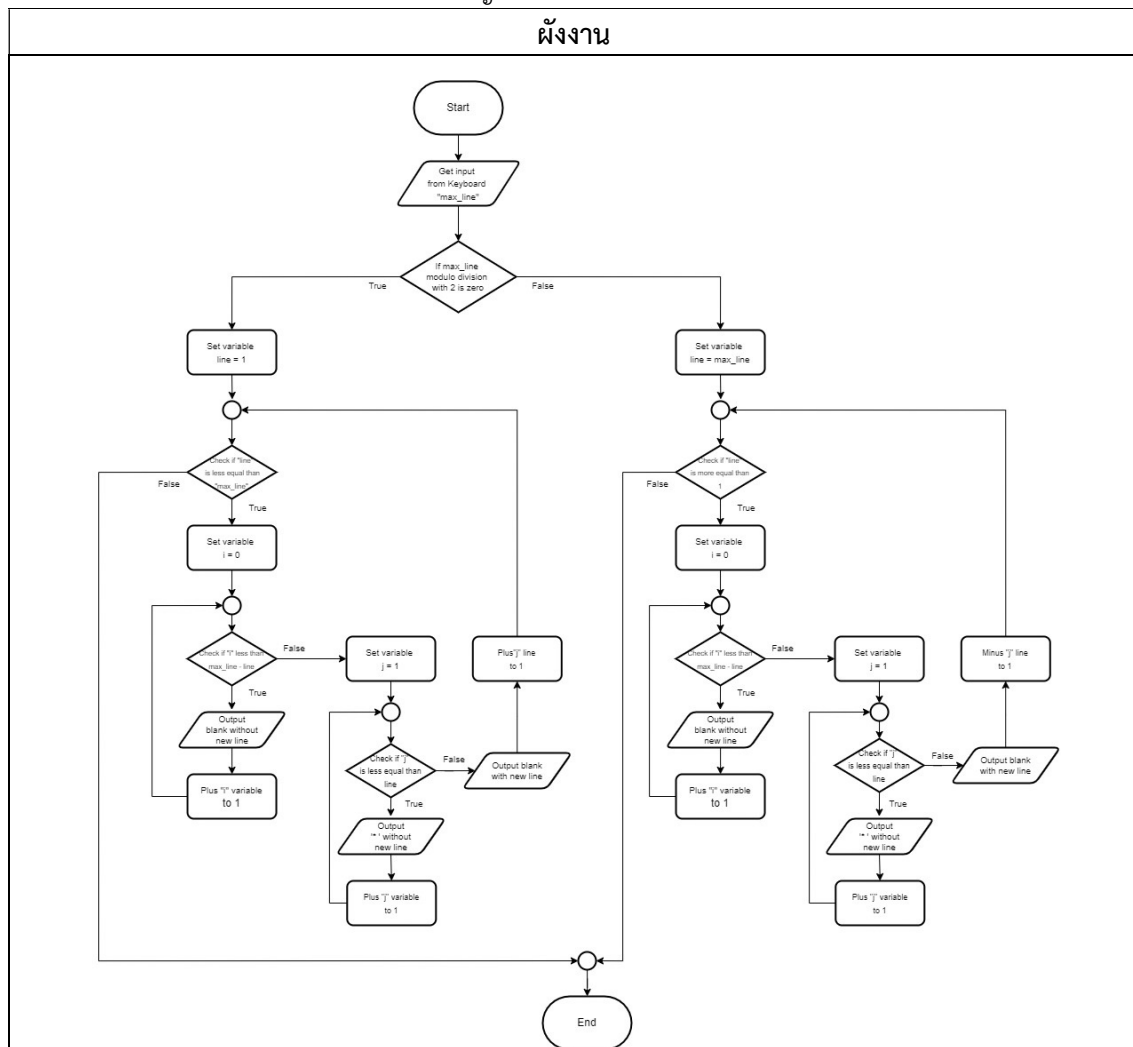
    for ( i = 1 ; i <= l ; i++){
        for ( j = 2 ; j <= i ; j++){
            printf( "  ");
        } //end for
        for ( j = 1 ; j >= i ; j--){
            printf ( " * ");
        } //end for
        printf ( "\n" );
    } // end for
    return 0;
} // end function
  
```

5.3. จงเขียนผังงานและโค้ดโปรแกรมเพื่อแก้ไขโจทย์ปัญหาดังต่อไปนี้

5.3.1. จงเขียนโปรแกรมเพื่อรับจำนวนบรรทัดจากผู้ใช้งาน และแสดงผลลัพธ์เป็นรูปต่อไปนี้ โดยหากเป็นเลขคี่ส่วนหัวของสามเหลี่ยมจะชี้ลง แต่หากเป็นเลขคู่หัวของสามเหลี่ยมจะชี้ขึ้น

Test Case		Test Case	
Input	Input your line : 5	Input	Input your line : 8
Output	<pre> * * * * * * * * * * * * * * * </pre>	Output	<pre> * * * * * * * * * * * * * * * * </pre>

5.3.2. ผังงานเพื่อแก้ไขโจทย์ปัญหา



5.3.3. โค้ดโปรแกรม

โค้ดโปรแกรม

```
#include <stdio.h>
int main() {
    int i = 0;
    int j = 0;
    int l = 0;

    printf("Enter Lines = ");
    scanf("%d", &l);

    if (l % 2 == 0) {
        for (i = 1; i <= l; i++) {
            for (j = 1; j >= i; j--) {
                printf(" ");
            } //end for
            for (j = 1; j <= i; j++) {
                printf("*");
            } //end for
            printf("\n");
        } //end for
    } else {
        for (i = 1; i <= l; i++) {
            for (j = 2; j <= i; j++) {
                printf(" ");
            } //end for
            for (j = 1; j >= i; j--) {
                printf("*");
            } //end for
            printf("\n");
        } //end for
    } //end if
    return 0;
} //end function
```

6. สรุปผลการปฏิบัติงาน

จากการทดลองจะเห็นว่า while loop จะวนซ้ำกับ for loop แต่ while loop จะวนจนจบได้
เร็วกว่า จนกว่าจะ break ดังนั้น

7. คำถามทางการทดลอง

7.1. จงระบุข้อผิดพลาดที่มักจะมีโอกาสเกิดขึ้นบ่อยในการใช้คำสั่ง for

การกำหนดเงื่อนไข = การผิดพลาดบ่อยๆ การกำหนดให้วนซ้ำครั้งเดียว
จะเกิดการทำงานโปรแกรม ไม่รันจนจบ loop ได้

7.2. จงระบุข้อควรระวังในการเขียนผังงานการวนรอบโดยใช้คำสั่ง for

ระวังการวัดเส้นเชื่อมโยงของ True และ False เพราะการวัดมีค่าแล้ว if else จะเกิด
ความสับสนได้