

ทักษะ

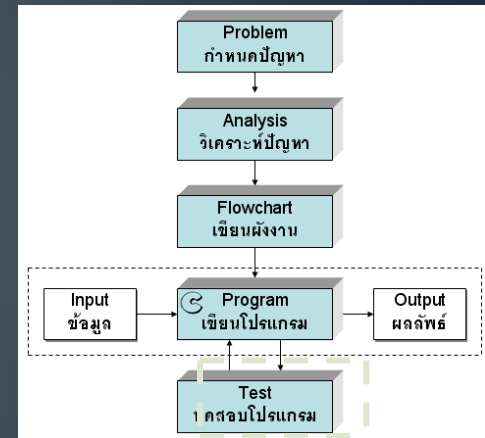
การโปรแกรมเบื้องต้น

การทดสอบและแก้ไขโปรแกรม

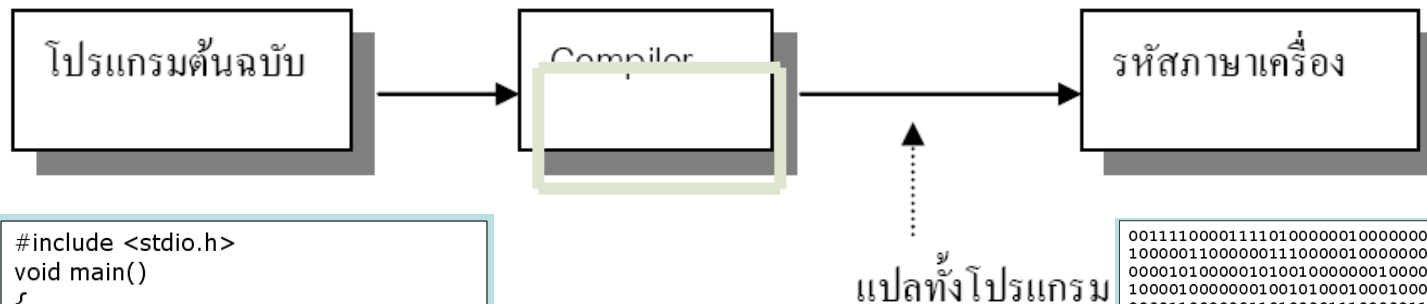
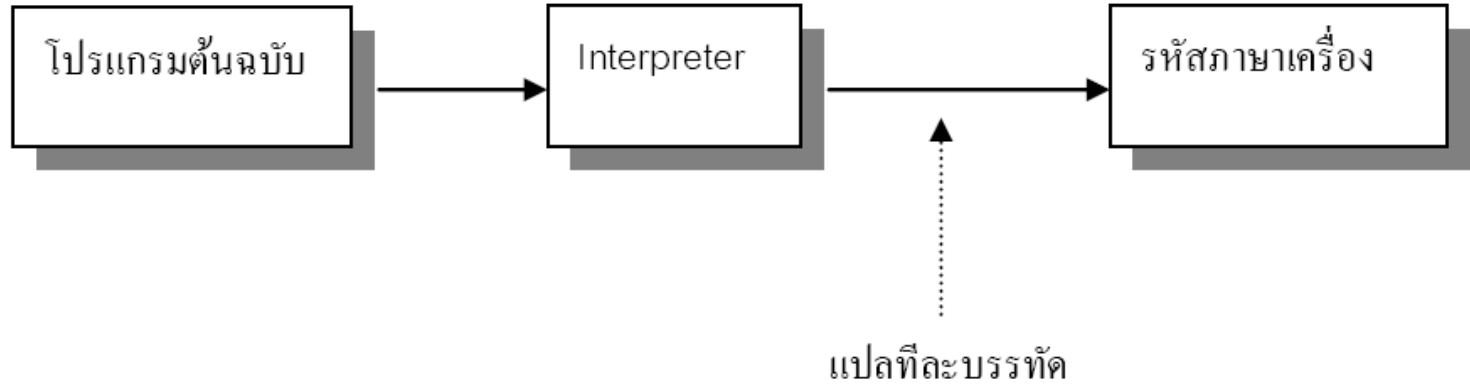
หลังจากเขียนโปรแกรมจะต้องทดสอบความถูกต้องของโปรแกรมที่เขียนขึ้น หากจุดผิดพลาดของโปรแกรมหามีหรือไม่ จุดผิดพลาดของโปรแกรมนี้นี้เรียกว่าบั๊ก (Bug) ส่วนการแก้ไขข้อผิดพลาดให้ถูกต้องเรียกว่า ดีบั๊ก(debug)

โดยทั่วไปแล้วข้อผิดพลาดจากการเขียนโปรแกรมจะมีสองประเภทคือ

1. การเขียนคำสั่งไม่ถูกต้องตามหลักการเขียนโปรแกรมภาษานั้น ๆ ซึ่งเรียกว่า Syntax Error หรือ Coding Error ข้อผิดพลาดประเภทนี้เรามักพบตอนแปลภาษาโปรแกรมเป็นรหัสภาษาเครื่อง
2. ข้อผิดพลาดทางตรรก หรือ Logic Error เป็นข้อผิดพลาดที่โปรแกรมทำงานได้ แต่ผลลัพธ์ออกมาไม่ถูกต้อง



ภาษาคอมพิวเตอร์และการพัฒนาโปรแกรม

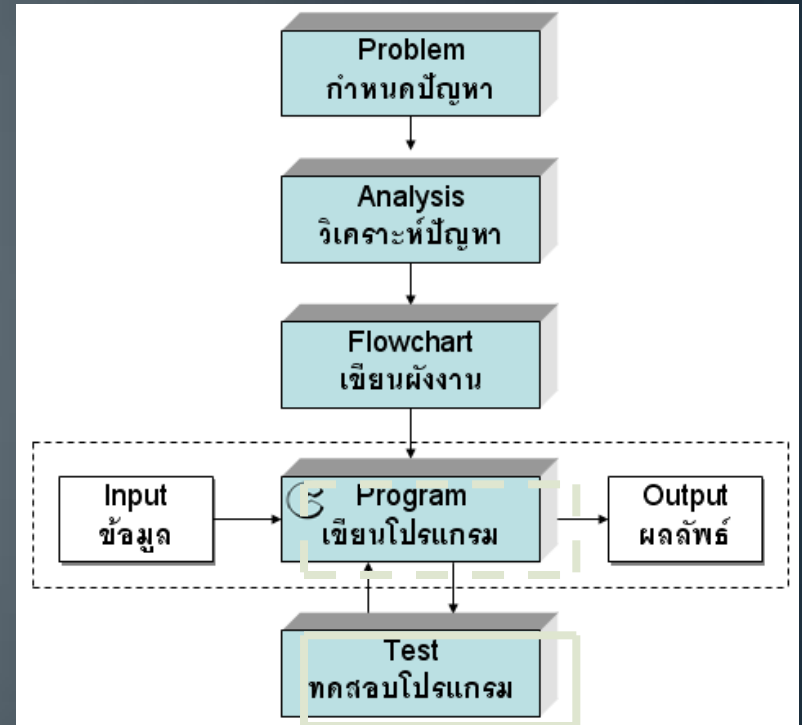


```
#include <stdio.h>
void main()
{
    int feet, inches;
    printf("Enter number (in feet) ");
    scanf("%d", &feet);
    inches = feet*12;
    printf(" = %d inches\n", inches);
}
```

```
00111100001111010000001000000011
10000011000000111000001000000010
00001010000010100100000001000010
10000100000001001010001000100011
00001100000011010000111000001111
10001001000010011000100000001000
00010100000101010001011000010111
00011000000110010001101000011011
00011100000111010001111000011111
00100100001001010010011000100111
00111100001111010011111000011111
00101000001010010010101000101011
00101100001011010010111000101111
00110000001100010011001000110011
10011011000110111001101000011010
00111000001110010011101000111011
```

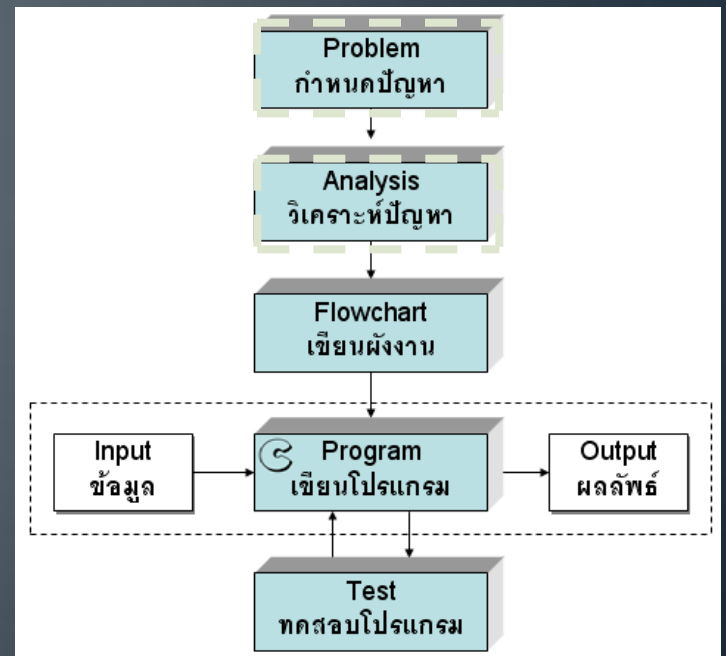
ภาษาคอมพิวเตอร์และการพัฒนาโปรแกรม

- กำหนดและวิเคราะห์ปัญหา
(Problem Definition and Problem Analysis)
- เขียนผังงานและซูโดโค้ด (Pseudocoding)
- เขียนโปรแกรม (Programming)
- ทดสอบและแก้ไขโปรแกรม
(Program Testing and Debugging)
- ทำเอกสารและบำรุงรักษาโปรแกรม
(Program Documentation and Maintenance)



การกำหนดและวิเคราะห์ปัญหา

- กำหนดขอบเขตของปัญหา
- กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ
(Input/Output Specification)
- กำหนดวิธีการประมวลผล
(Process Specification)



ตัวอย่าง

ตัวอย่าง

ถ้าหากต้องการออกแบบโปรแกรมให้คอมพิวเตอร์รับค่าข้อมูล 3 ค่า และแสดงค่าเฉลี่ยทางจอภาพ เราอาจกำหนดและวิเคราะห์ปัญหาได้ดังนี้

1. รับข้อมูลจากคีย์บอร์ด

- รับข้อมูลเฉพาะที่เป็นตัวเลขมาเก็บในตัวแปร (x_1, x_2, x_3)
- ถ้าข้อมูลเท่ากับ 0 ให้รับใหม่

ตัวอย่าง (ต่อ)

2. หาค่าเฉลี่ย

- รวมค่าทุกค่าที่รับมาเข้าด้วยกัน
- นำค่าผลรวมที่ได้หารด้วย 3
- นำค่าผลลัพธ์ไปเก็บในตัวแปร

$$10 + 20 + 30$$
$$(\text{sum} = x1 + x2 + x3)$$

$$(\text{mean} = \text{sum} / 3)$$

3. แสดงผลลัพธ์ทางจอภาพ

- แสดงคำว่าค่าเฉลี่ยเท่ากับ
- แสดงผลลัพธ์โดยมีทศนิยมสองตำแหน่ง (Mean = \uparrow)
20.00

ขั้นตอนการทำงาน

นำข้อมูลเข้า

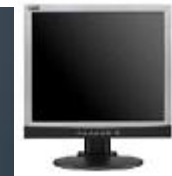
รับข้อมูลตัวที่ 1
 $x_1 = 10$
รับข้อมูลตัวที่ 2
 $x_2 = 20$
รับข้อมูลตัวที่ 3
 $x_3 = 30$

ประมวลผล

อ่านค่าเฉพาะที่เป็นเลข 3 ตัว
นำตัวเลขทั้งสามตัวมารวมกัน
($sum = x_1 + x_2 + x_3$)
นำผลรวมมาหารด้วย 3
($mean = sum / 3$)

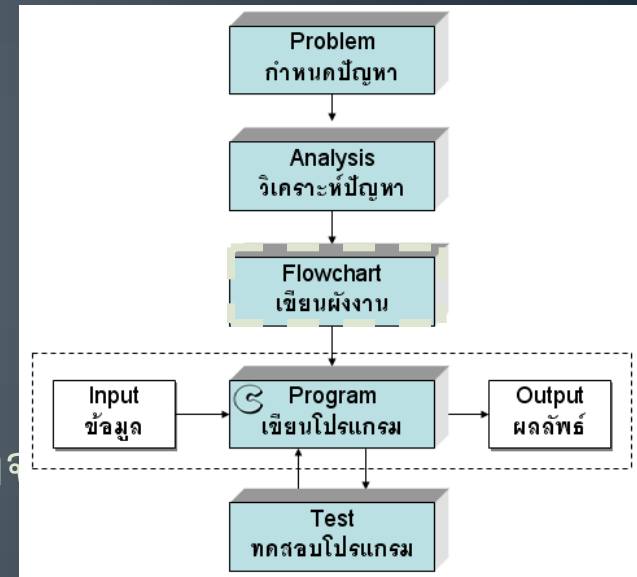
นำข้อมูลออก

แสดงค่าเฉลี่ย
ทศนิยมสอง
ตำแหน่ง
Mean=20.00



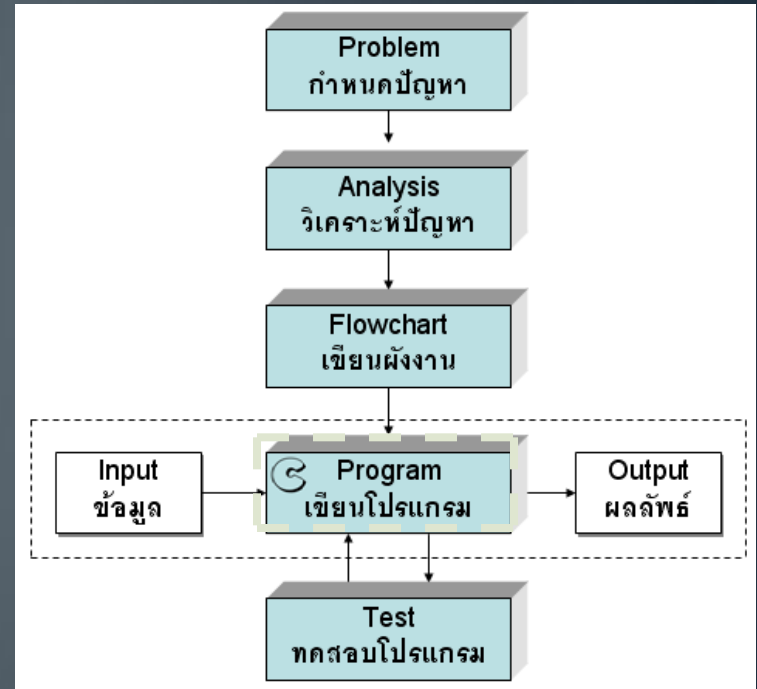
การเขียนผังงานและซูโดโค้ด

หลังจากที่ได้วิเคราะห์ปัญหาแล้ว ขั้นตอนต่อไปจะใช้เครื่องมือช่วยในการออกแบบโปรแกรม โดยเขียนเป็นลำดับขั้นตอนการทำงานของโปรแกรมที่เรียกว่า อัลกอริทึม (Algorithm) โดยจะเขียนแสดงขั้นตอนการแก้ปัญหา ด้วยประโยคที่ชัดเจนไม่คลุมเครือ และมีรายละเอียดการทำงานพอสมควรเพียงพอที่จะนำไปเขียนเป็นโปรแกรมให้ทำงานจริง โดยอัลกอริทึมนั้นอาจเขียนให้อยู่ในรูปของรหัสจำลองหรือซูโดโค้ด (Pseudo-code) หรือเขียนเป็นผังงาน (Flowchart) ก็ได้



การเขียนโปรแกรม

เปลี่ยนขั้นตอนการทำงานให้อยู่ในรูป
รหัสภาษาคอมพิวเตอร์ การเขียนโปรแกรม
จะต้องเขียนตามภาษาที่คอมพิวเตอร์เข้าใจ
โดยอาจใช้ภาษาระดับสูงหรือระดับต่ำซึ่ง
สามารถเลือกได้หลายภาษา การเขียน
โปรแกรมแต่ละภาษาจะต้องทำตามหลัก
ไวยากรณ์ (syntax) ที่กำหนดไว้ในภาษานั้น



ทำเอกสารและบำรุงรักษาโปรแกรม

ขั้นตอนนี้จะทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างมีประสิทธิภาพ และสะดวกในการตรวจสอบข้อผิดพลาดโดยเขียนเป็นเอกสารประกอบโปรแกรมขึ้นมา โดยทั่วไปแล้วแบ่งออกเป็นสองประเภทคือ

1. คู่มือการใช้ หรือ User Document หรือ User guide ซึ่งจะอธิบายการใช้โปรแกรม
2. คู่มือโปรแกรมเมอร์ หรือ Program Document หรือ Technical Reference ซึ่งจะอำนวยความสะดวกในการแก้ไขโปรแกรมและพัฒนาโปรแกรมในอนาคต โดยจะมีรายละเอียดต่างๆ เกี่ยวกับโปรแกรม เช่นชื่อโปรแกรม การรับข้อมูล การพิมพ์ผลลัพธ์ ขั้นตอนต่าง ๆ ในโปรแกรม เป็นต้น

ส่วนการบำรุงรักษาโปรแกรม (Maintenance) เป็นการที่ผู้เขียนโปรแกรมจะต้องคอยตรวจสอบการใช้โปรแกรมจริง เพื่อแก้ไขข้อผิดพลาดซึ่งอาจเกิดขึ้นในภายหลัง รวมทั้งพัฒนาโปรแกรมให้ทันสมัยอยู่เสมอเมื่อเวลาผ่านไป

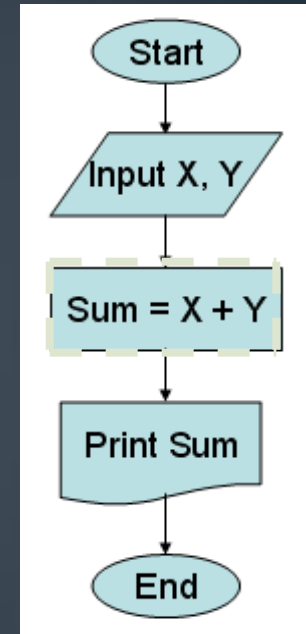
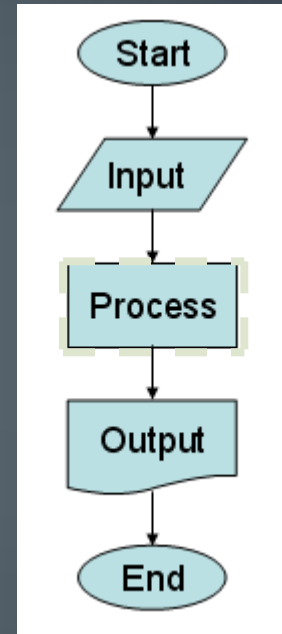
กระบวนการแก้ปัญหาด้วยคอมพิวเตอร์

วิเคราะห์ปัญหา

- ต้องการอะไร (ค่าเฉลี่ยของข้อมูล 3 ค่า)
- ต้องการเอาต์พุตอย่างไร (Mean = ?)
- ข้อมูลเข้าเป็นอย่างไร (10, 20, 30)
- ตัวแปรที่ใช้ (x1, x2, x3)
- วิธีการประมวลผลเป็นอย่างไร (sum = x1 + x2 + x3)
(mean = sum / 3)

การเขียนผังงาน

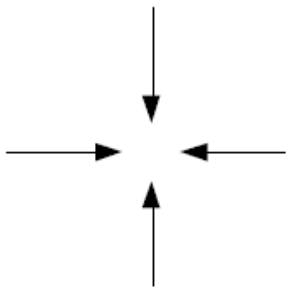
ผังงานเป็นแผนภาพที่แสดงลำดับขั้นตอนในการทำงานของโปรแกรม โดยมีการลงรายละเอียด ใส่วิธีการ และจัดลำดับขั้นตอนของโปรแกรม ตั้งแต่เริ่มต้นจากการรับข้อมูล การประมวลผล ไปจนถึงการแสดงผลการทำงานของ



สัญลักษณ์ที่ใช้ในการเขียนผังงาน



การเริ่มต้นและการสิ้นสุดการทำงานของโปรแกรม



ลูกศรแสดงทิศทางการทำงานของโปรแกรม และการไหลของข้อมูล

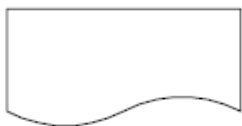
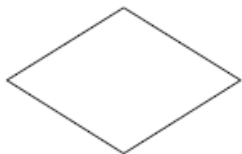


การประมวลผลการคำนวณต่าง ๆ



การทำงานย่อย

สัญลักษณ์ที่ใช้ในการเขียนผังงาน



การรับหรือแสดงข้อมูล โดยไม่ระบุชนิดของ
อุปกรณ์

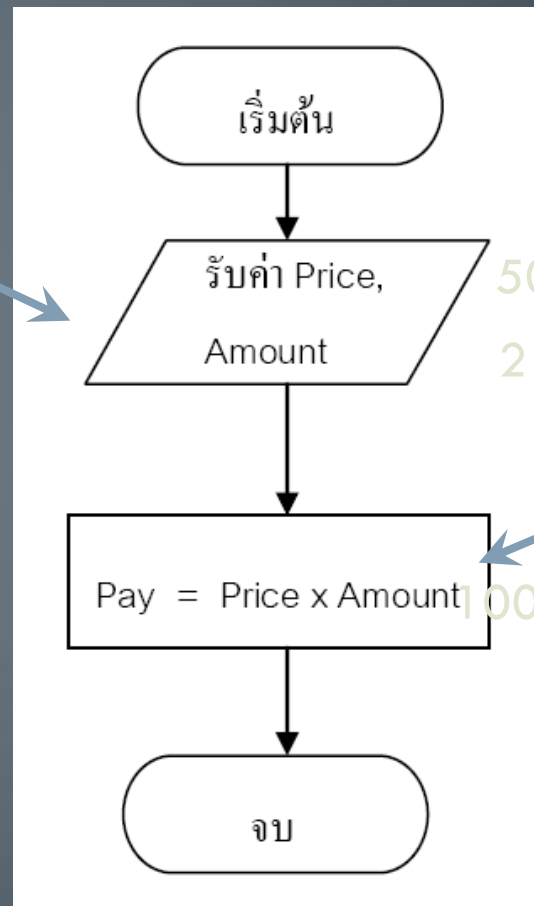
การตรวจสอบเงื่อนไข เพื่อเลือกทำอย่างใด
อย่างหนึ่ง โดยจะมีเส้นลูกศรแสดงทิศ
ทางการทำงานต่อไป

แสดงผลลัพธ์หรือรายงานทางเครื่องพิมพ์

จุดเชื่อมต่อของผังงาน

ตัวอย่างการเขียนผังงาน

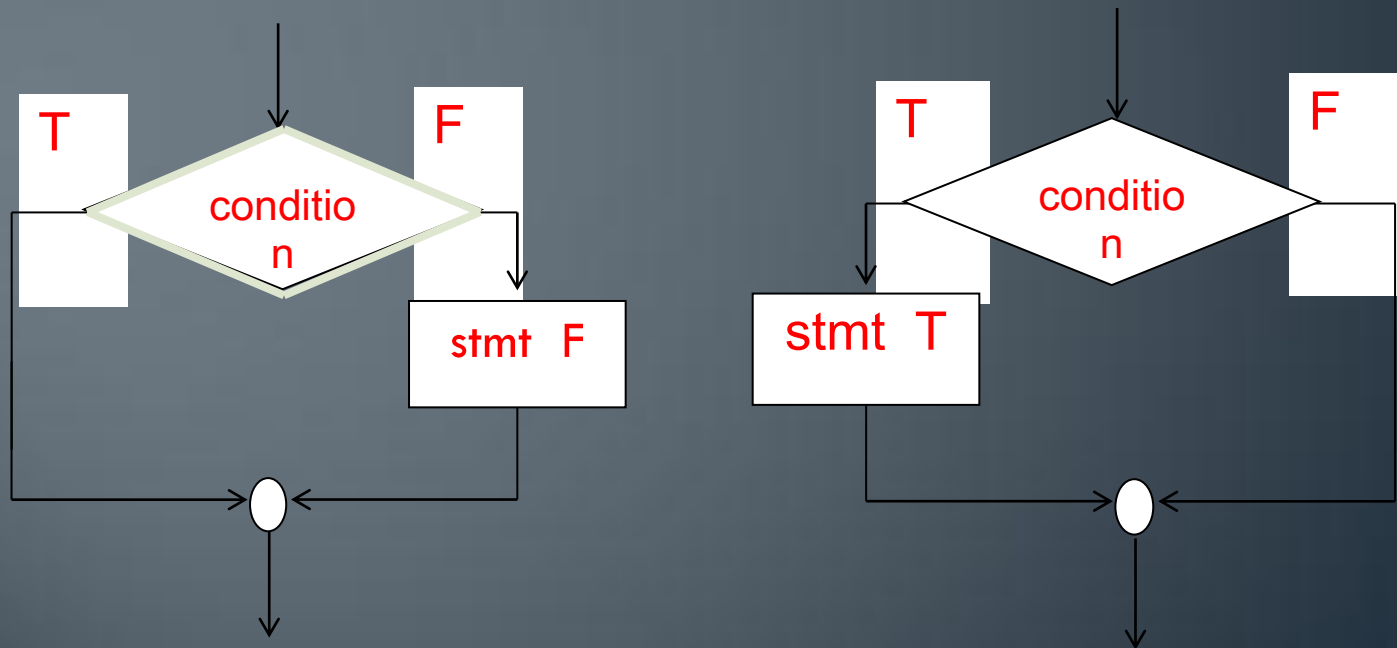
สัญลักษณ์ของ
การรับค่า



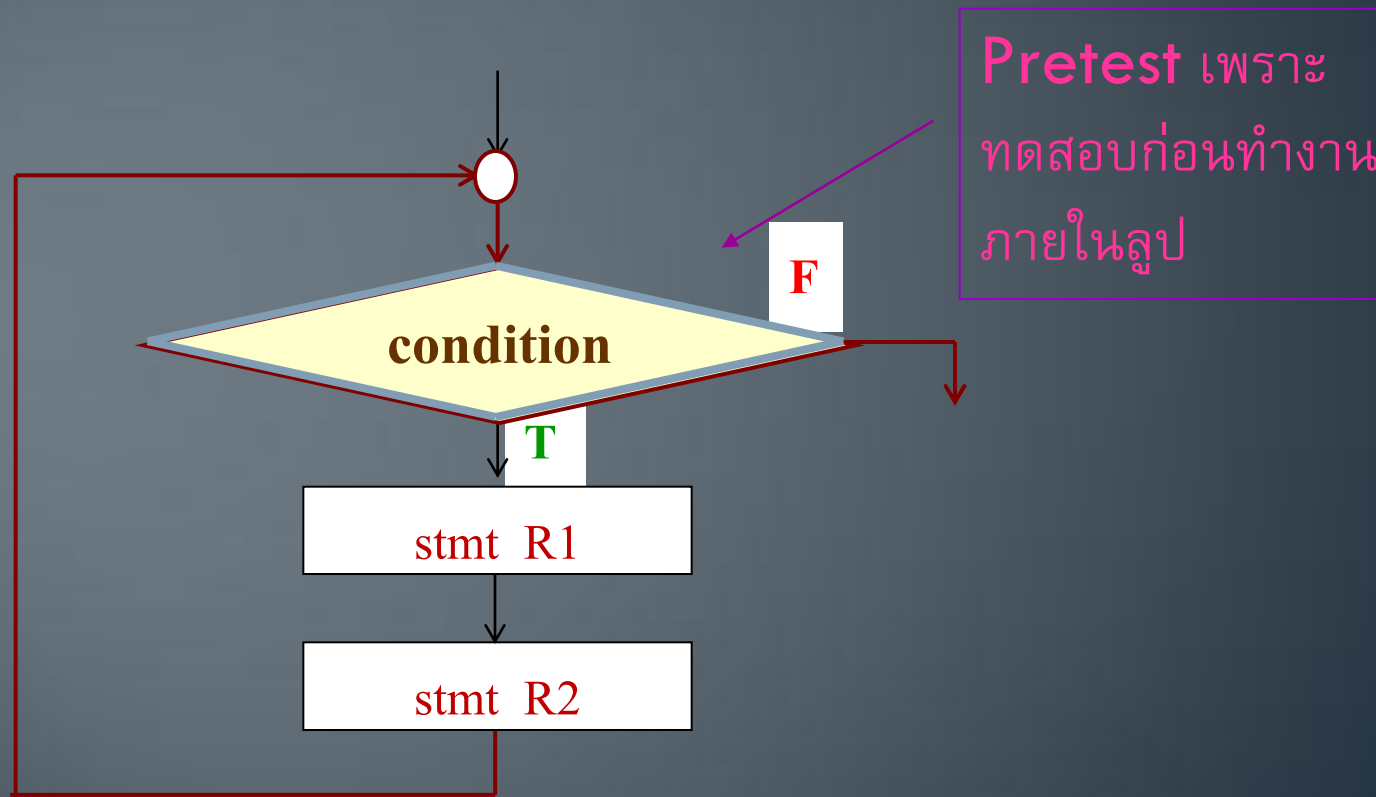
สัญลักษณ์ของ
การคำนวณ

ผังงานแบบทางเลือก

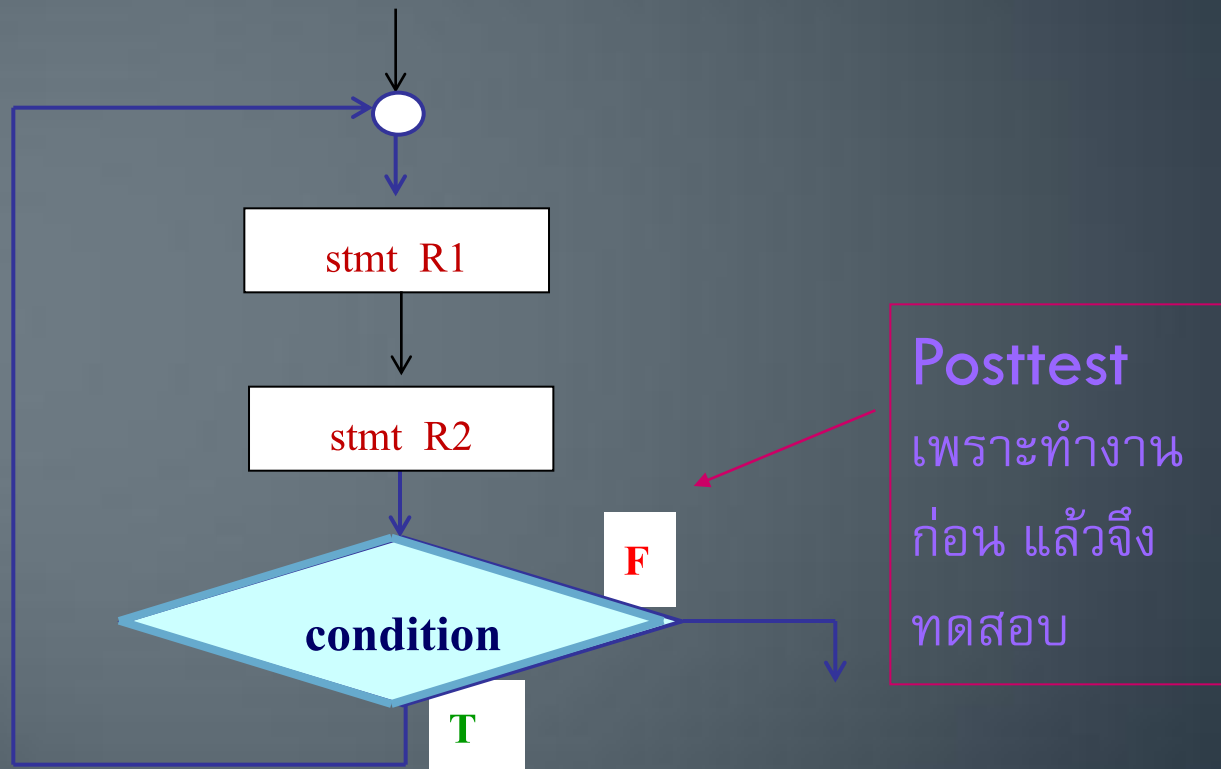
เป็นการทำงานที่ต้องมีการตัดสินใจ เพื่อเลือกการทำงาน โดยใช้สัญลักษณ์การเปรียบเทียบ



ผังงานแบบวนซ้ำ (while-do)



ผังงานแบบวนซ้ำ (do-while)

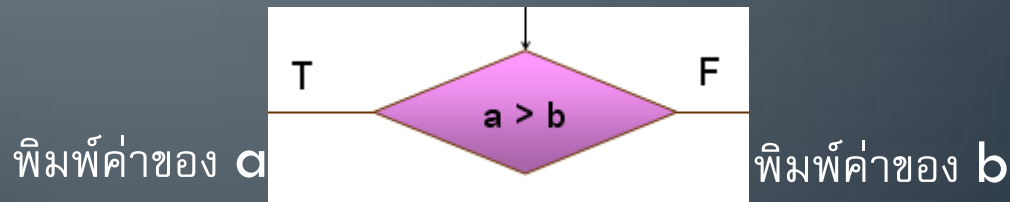


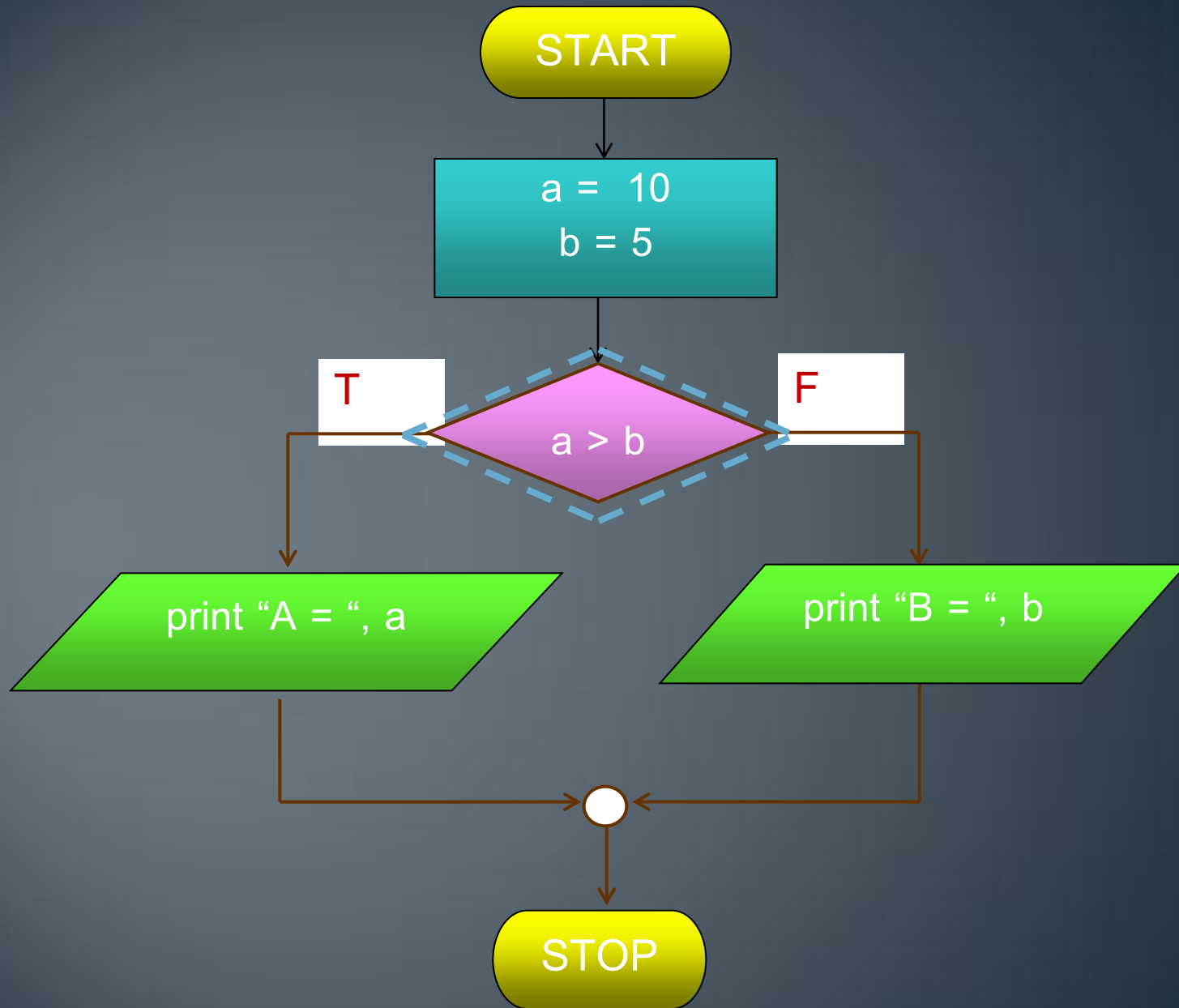
ตัวอย่าง

ตัวอย่าง จงเขียนผังงานเพื่อกำหนดค่าให้กับตัวแปร **a** และ **b** แล้วเปรียบเทียบเพื่อแสดงผล

ถ้าตัวแปร **a** มีค่ามากกว่าตัวแปร **b** ให้พิมพ์ค่าตัวแปร **a** ออกมาแสดงบนจอภาพ

ถ้าไม่ใช่ ให้พิมพ์ค่าตัวแปร **b** ออกมาแสดงบนจอภาพ





1.3 การเขียนผังงาน

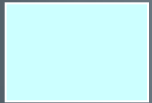
ผังงาน (Flowchart) เป็นแผนภาพ ที่ใช้อธิบายขั้นตอนและลำดับการทำงานของโปรแกรมตั้งแต่เริ่มจนจบ

- สัญลักษณ์พื้นฐาน ที่ใช้ในผังงาน

 แสดงจุดเริ่มต้น หรือจุดจบของผังงาน



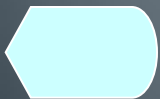
แสดงทิศทางของการดำเนินงาน



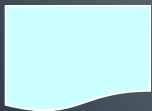
แสดงการคำนวณ



แสดงการรับข้อมูล หรือแสดงผล (ไม่ระบุชนิดอุปกรณ์)



การแสดงผลทางจอภาพ



การแสดงผลทางเครื่องพิมพ์



แสดงการทดสอบเงื่อนไข (เมื่อมีทางเลือก)



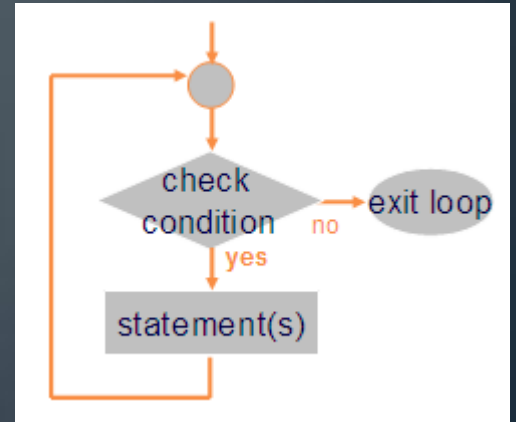
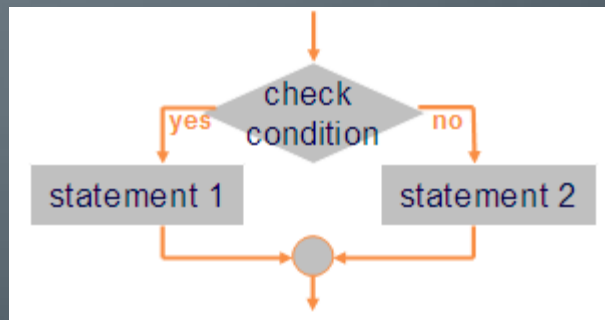
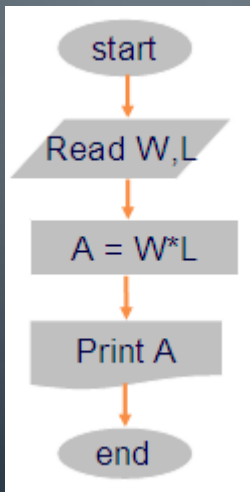
แสดงจุดเชื่อมต่อในผังงาน

รูปแบบของผังงาน

● ผังงานแบบลำดับ

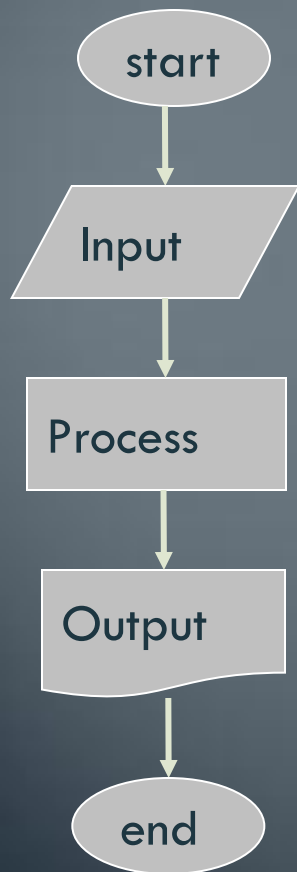
● ผังงานแบบเงื่อนไข

● ผังงานแบบวนซ้ำ



1.3.1 ผังงานแบบลำดับ

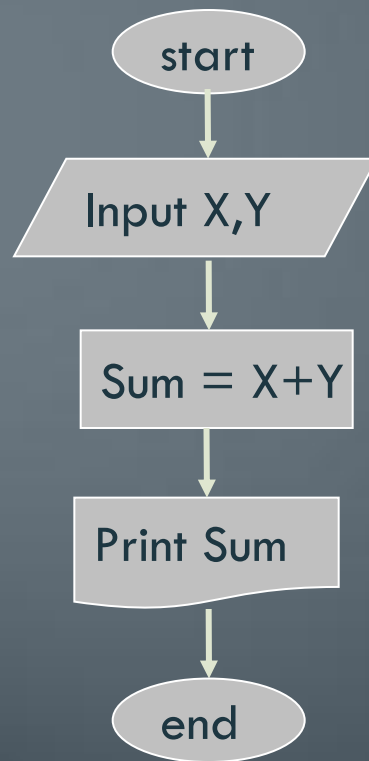
- Flowchart แบบลำดับ (Sequence)



แสดงขั้นตอนการทำงานที่ทำงานตามลำดับ (จากบนลงล่าง) ไม่มีการข้ามขั้น หรือ ย้อนกลับ)

ตัวอย่าง 1.4

- แสดง **Flowchart** ของการหาผลบวก (Sum) ของข้อมูล 2 ค่า (X, Y) พร้อมแสดงผลบวก



	Memory
X	50
Y	10
Sum	10
	0
	...

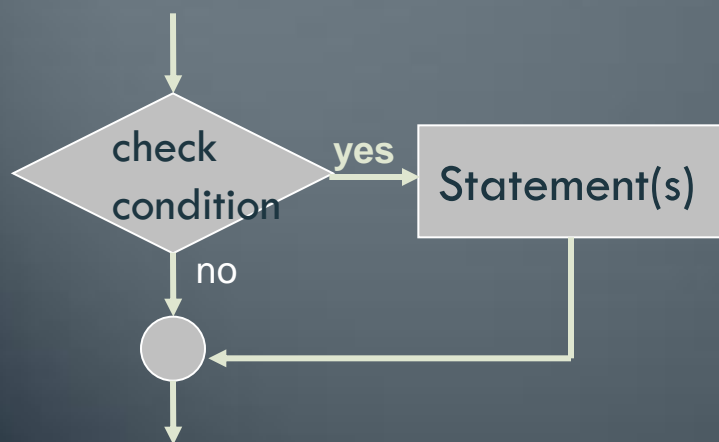
1.3.2 ผังงานแบบเงื่อนไข

แสดงการทำงานตามเงื่อนไข

ที่จะต้องเลือกทำอย่างใดอย่างหนึ่ง

ซึ่งมี 3 กรณี คือ เลือกจาก 1, 2, หรือ n

1. เงื่อนไขทางเลือกจาก 1 เส้นทาง



จะทำงาน (Statement) เมื่อ
เงื่อนไขเป็นจริงเท่านั้น

ตัวอย่าง 1.5

- แสดง **Flowchart** เพื่อตรวจสอบเงื่อนไข
เพื่อคัดเลือกที่สอบผ่าน (คะแนน $X \geq 60$)
และแสดงผลเฉพาะผู้ที่สอบผ่าน

