

## หน่วยที่ 6 คำสั่งควบคุม

### คำสั่งควบคุมแบบทางเลือก (Selection)

เป็นโครงสร้างการเขียนโปรแกรมที่มีการทดสอบเงื่อนไข เพื่อตัดสินใจทำงานอย่างใดอย่างหนึ่ง โดยจะมีการข้ามไม่ทำงานบางคำสั่ง มีอยู่ 3 รูปแบบ ดังนี้

1. การเลือกทำแบบหนึ่งเส้นทาง (if)
2. การเลือกทำแบบสองเส้นทาง (if-else)
3. การเลือกทำแบบหลายเส้นทาง (if-elif-else, Nested if, match-case)

#### 1. การเลือกทำแบบหนึ่งเส้นทาง (if)

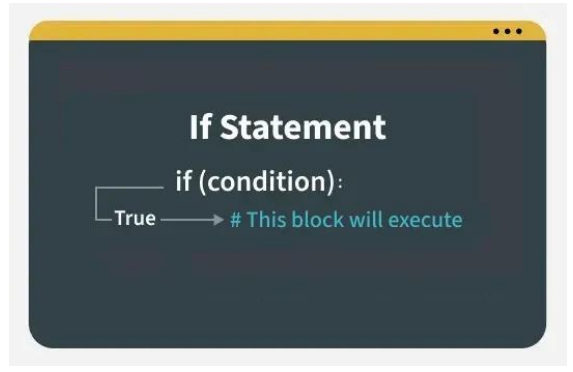
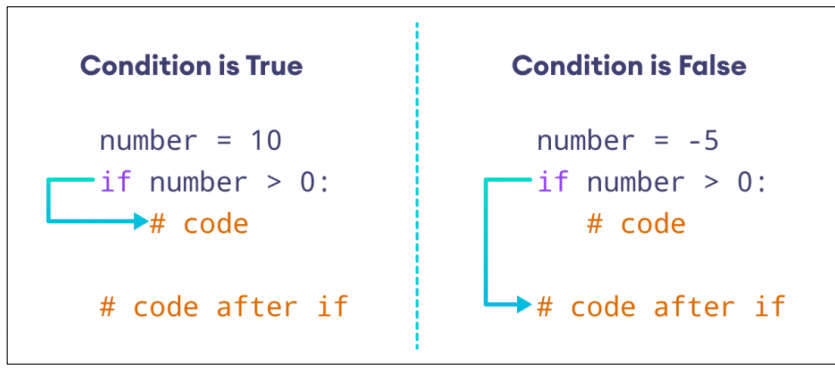
รูปแบบคำสั่ง

```
if condition :
    statement
```

หรือ

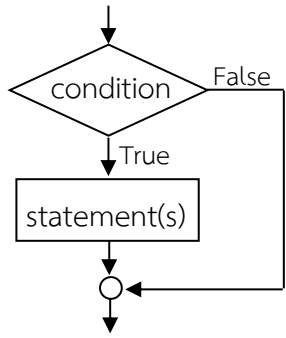
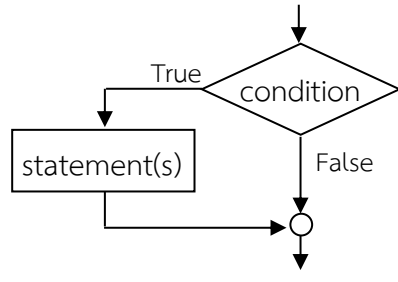
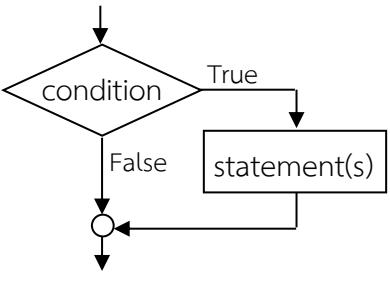
```
if condition :
    statement_1
    statement_2
    ...
    statement_n
```

ถ้า...แล้ว...



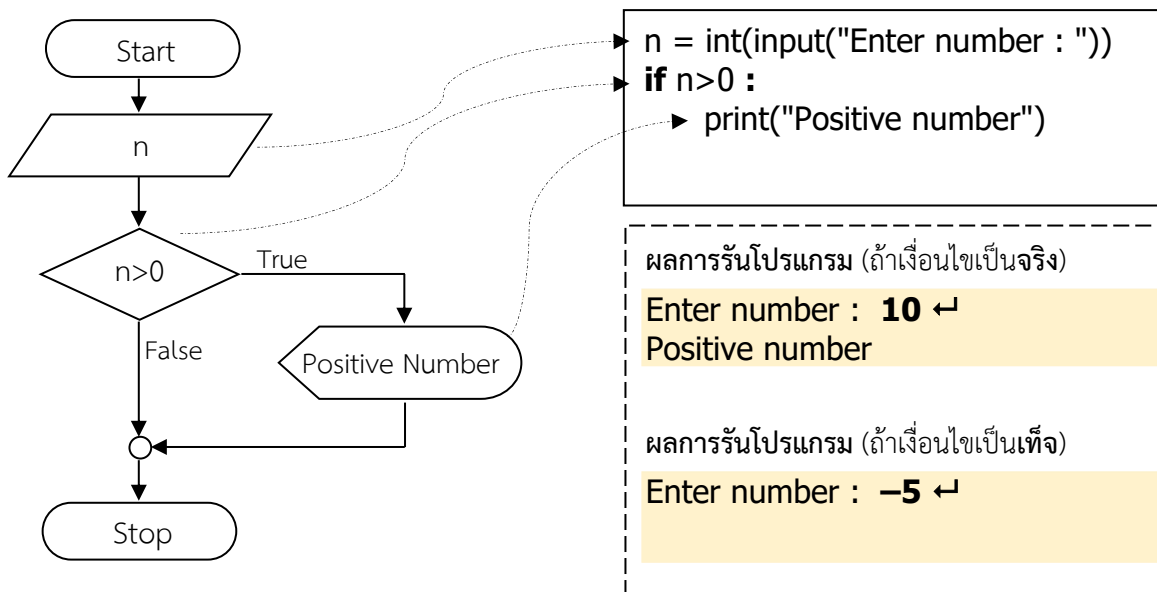
**คำอธิบาย condition** คือ เงื่อนไขที่กำหนดขึ้น เพื่อใช้ตัดสินใจว่าจะทำงานหรือไม่ทำงานตามคำสั่ง โดยเงื่อนไขจะต้องเขียนอยู่หลังคำสั่ง **if** ซึ่งเงื่อนไขอาจอยู่ในรูปของนิพจน์ (Expression) การคำนวณเปรียบเทียบ หรือเป็นคำสั่งของตัวแปรก็ได้ ที่จะมีการเปรียบเทียบเงื่อนไขว่าเป็นจริง (True) หรือเป็นเท็จ (False)

**statement(s)** คือ คำสั่งที่จะให้ทำงานถ้าผลการตรวจสอบเงื่อนไข (condition) ออกมาเป็นจริง (True) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในย่อหน้าเดียวกัน

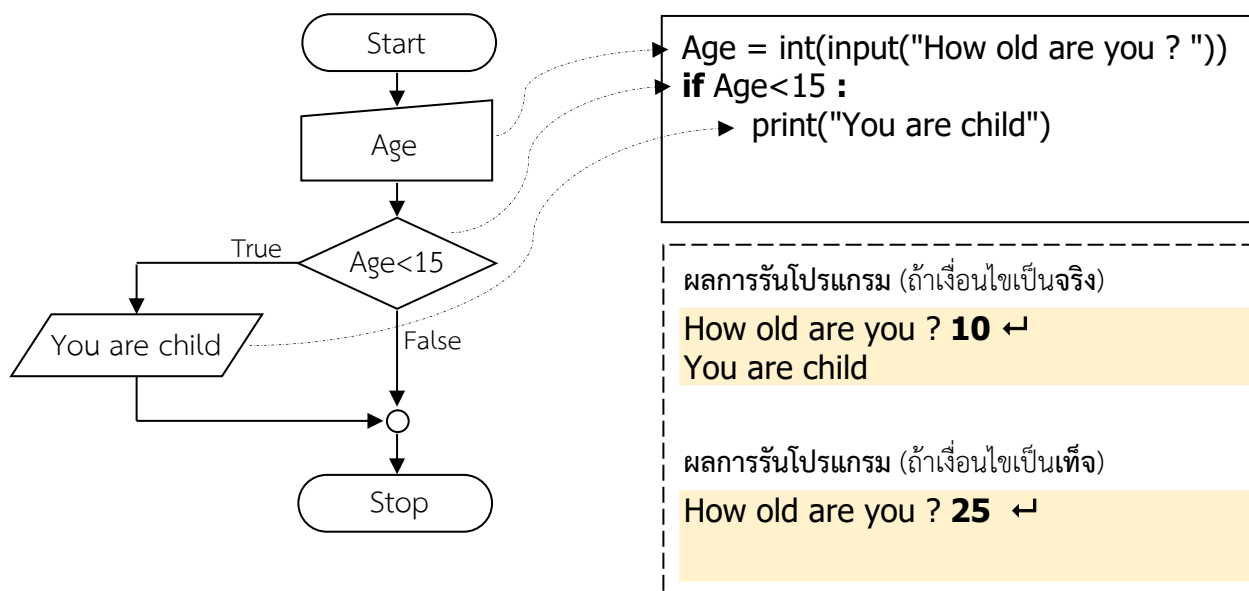


## ตัวอย่างโปรแกรมคำสั่งการเลือกทำแบบหนึ่งเส้นทาง (if)

Ex1. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบหาค่า ถ้าตัวเลขมากกว่าศูนย์ แล้วแสดงผลลัพธ์เป็นจำนวนเต็มบวก (Positive Number)



Ex2. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มที่เป็นอายุ ถ้าอายุน้อยกว่า 15 ปี แล้วแสดงข้อความคุณเป็นเด็ก (You are child)



## 2. การเลือกทำแบบสองเส้นทาง (if-else)

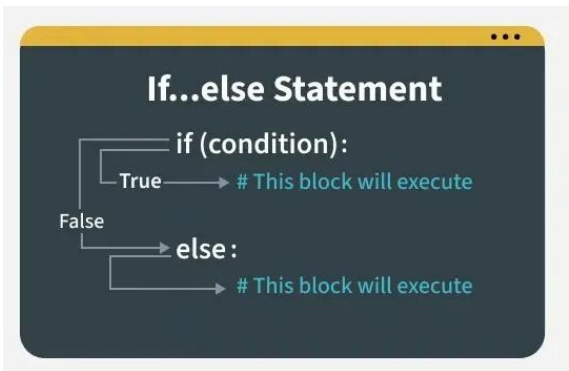
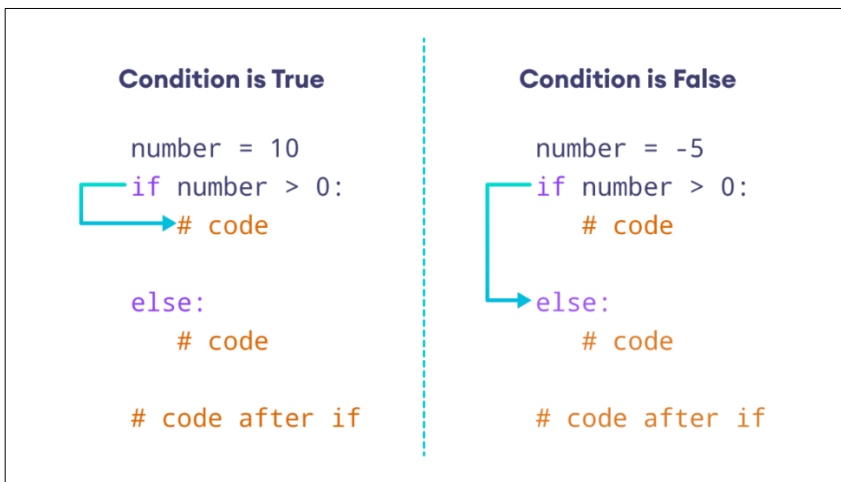
รูปแบบคำสั่ง

ถ้า...แล้ว...มิฉะนั้น...

```
if condition :
    statement1
else :
    statement2
```

หรือ

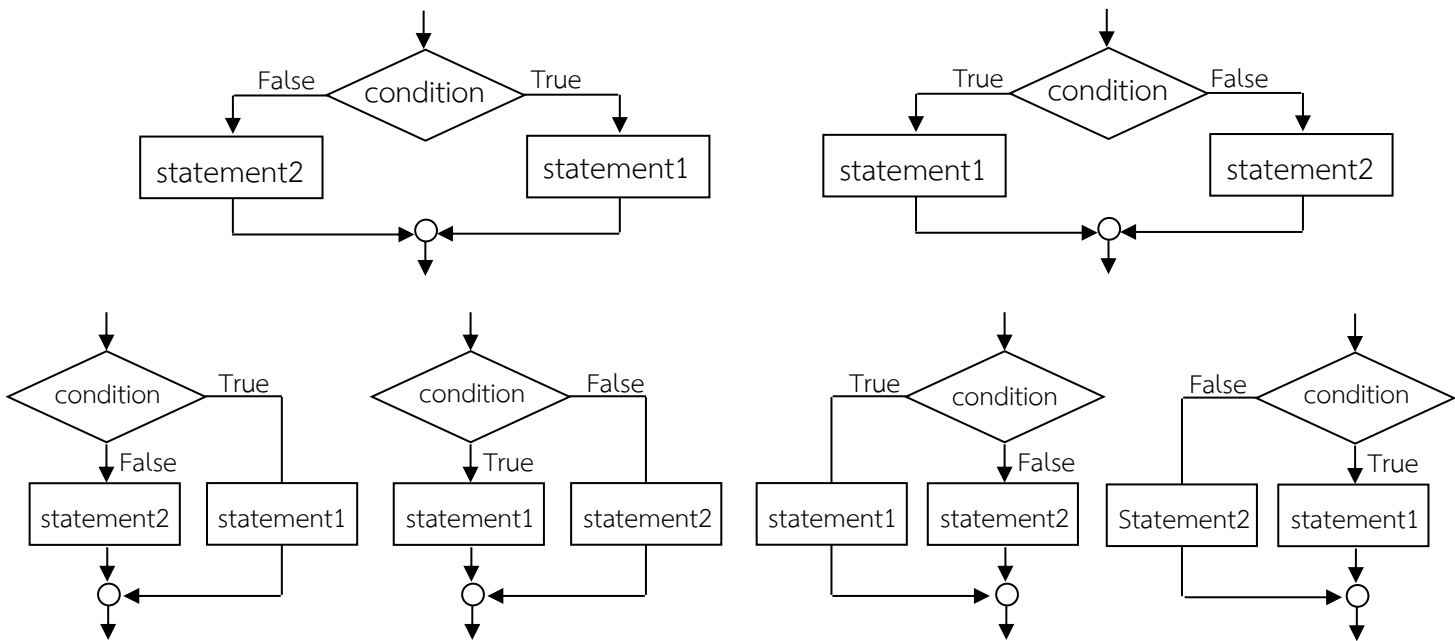
```
if condition :
    statement1
    ...
    statement1_n
else :
    statement2;
    ...
    statement2_n
```



**คำอธิบาย condition** คือ เงื่อนไขที่กำหนดขึ้น เพื่อใช้ตัดสินใจว่าจะทำงานหรือไม่ทำงานตามคำสั่ง โดยเงื่อนไขจะต้องเขียนอยู่หลังคำสั่ง if ซึ่งเงื่อนไขอาจอยู่ในรูปของนิพจน์ (Expression) การคำนวณเปรียบเทียบ หรือเป็นคำสั่งของตัวแปรก็ได้ ที่จะมีการเปรียบเทียบเงื่อนไขว่าเป็นจริง (True) หรือเป็นเท็จ (False)

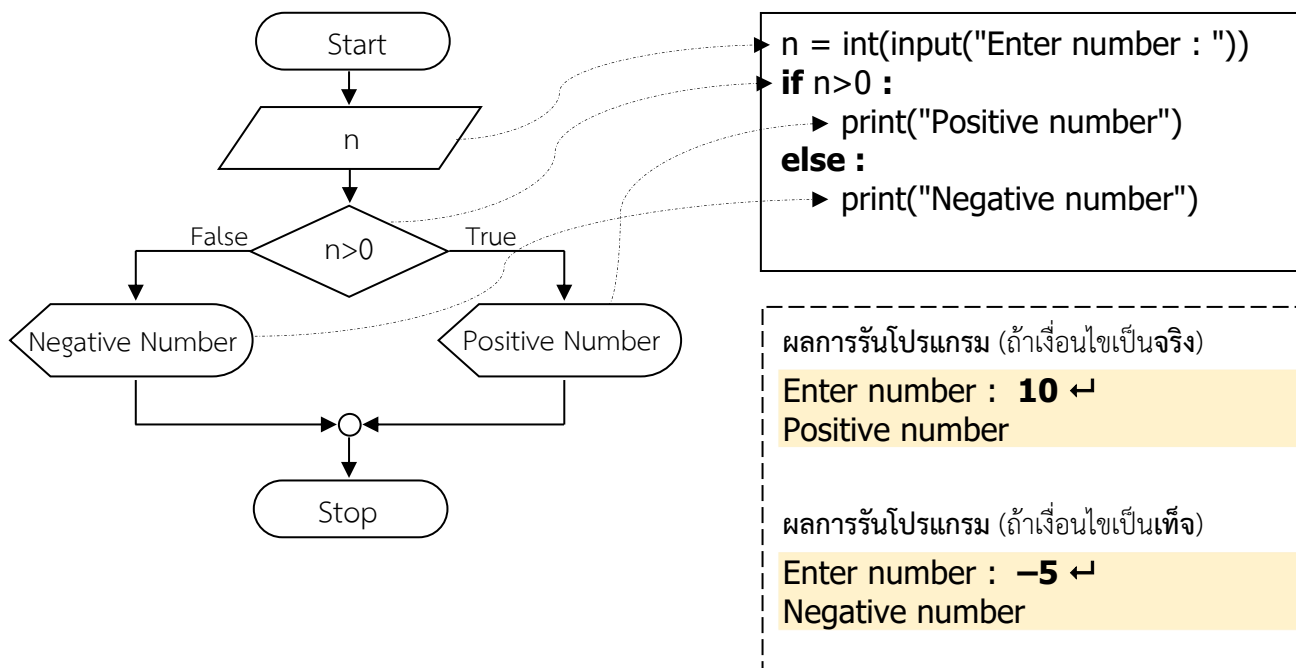
**statement1** คือ คำสั่งที่จะให้ทำงานถ้าผลการตรวจสอบเงื่อนไข (condition) ออกมาเป็นจริง (True) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในย่อหน้าเดียวกัน

**statement2** คือ คำสั่งที่จะให้ทำงานถ้าผลการตรวจสอบเงื่อนไข (condition) ออกมาเป็นเท็จ (False) ซึ่งอาจจะมีคำสั่งที่ต้องการให้ทำงานมากกว่าหนึ่งคำสั่งก็ได้ แต่ต้องเขียนคำสั่งเหล่านั้นไว้ภายในย่อหน้าเดียวกัน

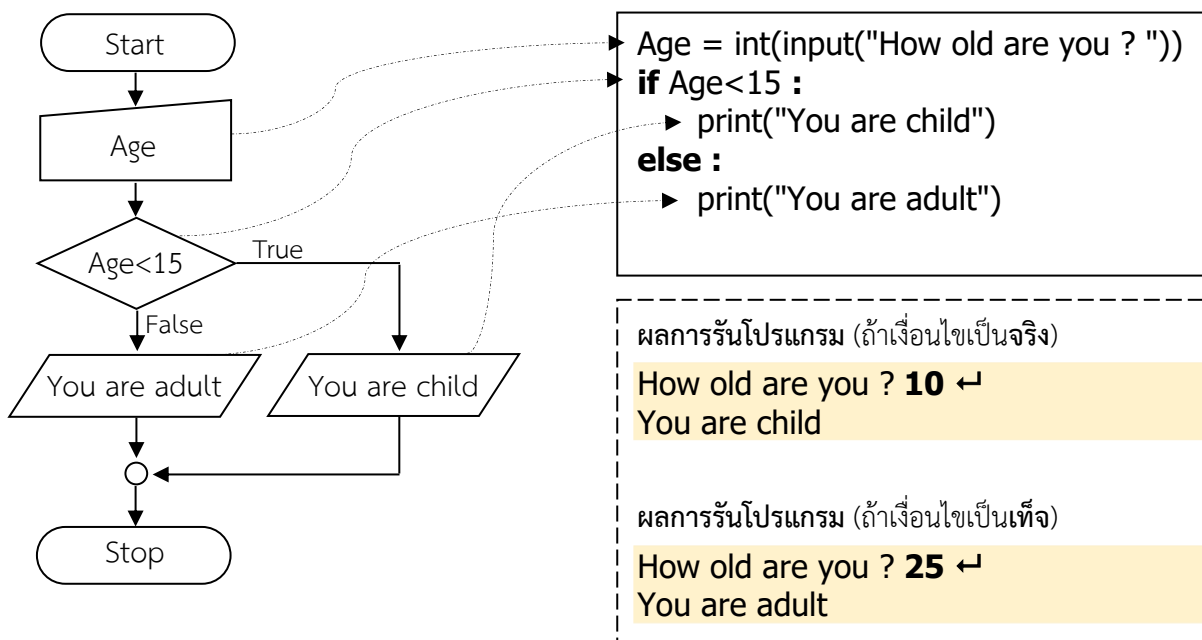


### ตัวอย่างโปรแกรมคำสั่งการเลือกทำแบบสองเส้นทาง (if-else)

Ex3. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบหาค่า ถ้าตัวเลขมากกว่าศูนย์ แล้วแสดงผลเป็นจำนวนเต็มบวก (Positive Number) มิฉะนั้นแสดงผลเป็นจำนวนเต็มลบ (Negative Number)



Ex4. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มที่เป็นอายุ ถ้าอายุน้อยกว่า 15 ปี แล้วแสดงข้อความคุณเป็นเด็ก (You are child) มิฉะนั้นแสดงข้อความ คุณเป็นผู้ใหญ่ (You are adult)



### 3. การเลือกทำแบบหลายเส้นทาง (if-elif-else, Nested if และ match-case)

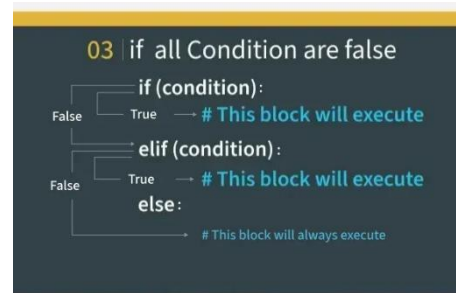
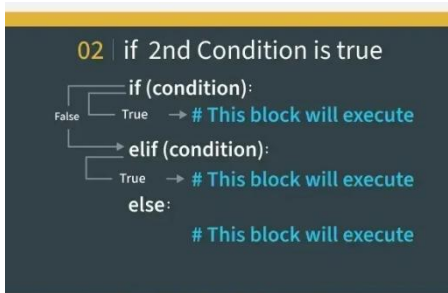
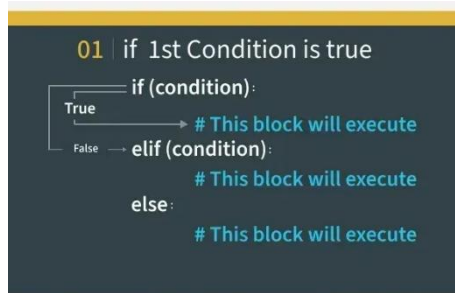
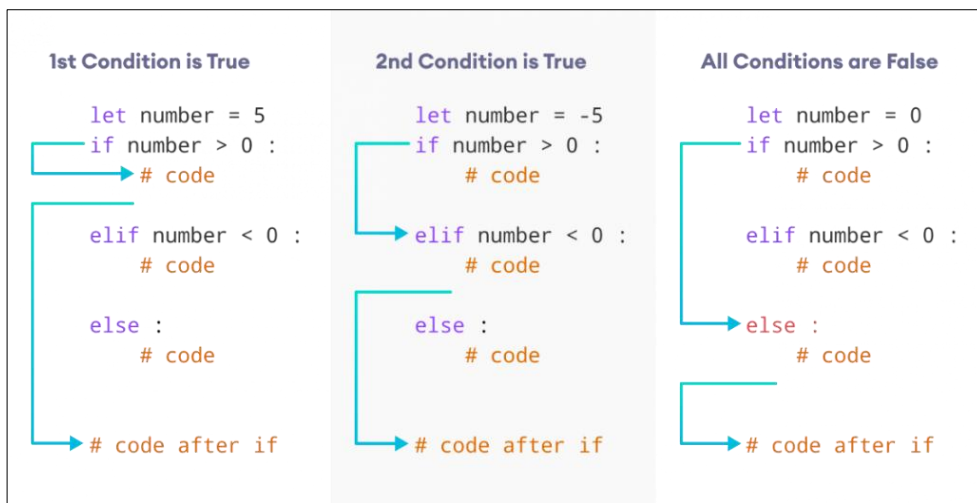
#### 3.1 การเลือกทำแบบหลายเส้นทางแบบ if-elif-else

รูปแบบคำสั่ง

```

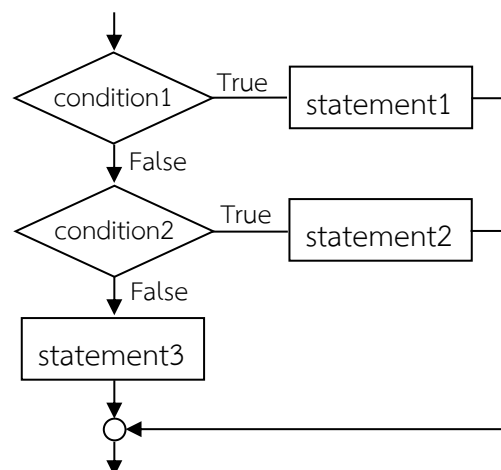
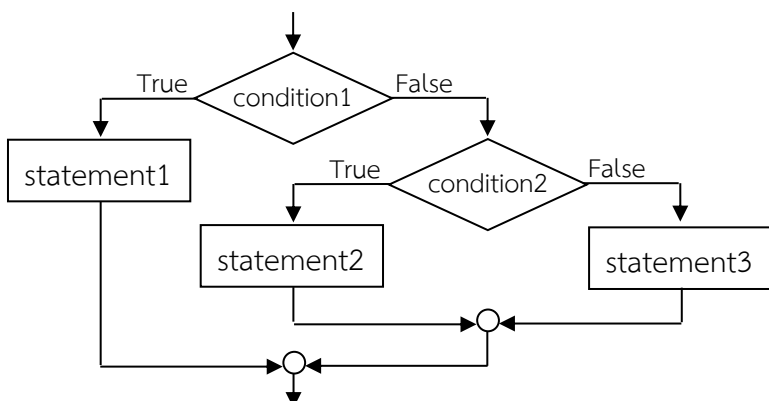
if condition1 :
    statement1
elif condition2 :
    statement2
elif condition3 :
    statement3
    .....
    .....
elif condition_n-1 :
    statement_n-1
else :
    statement_n
    
```

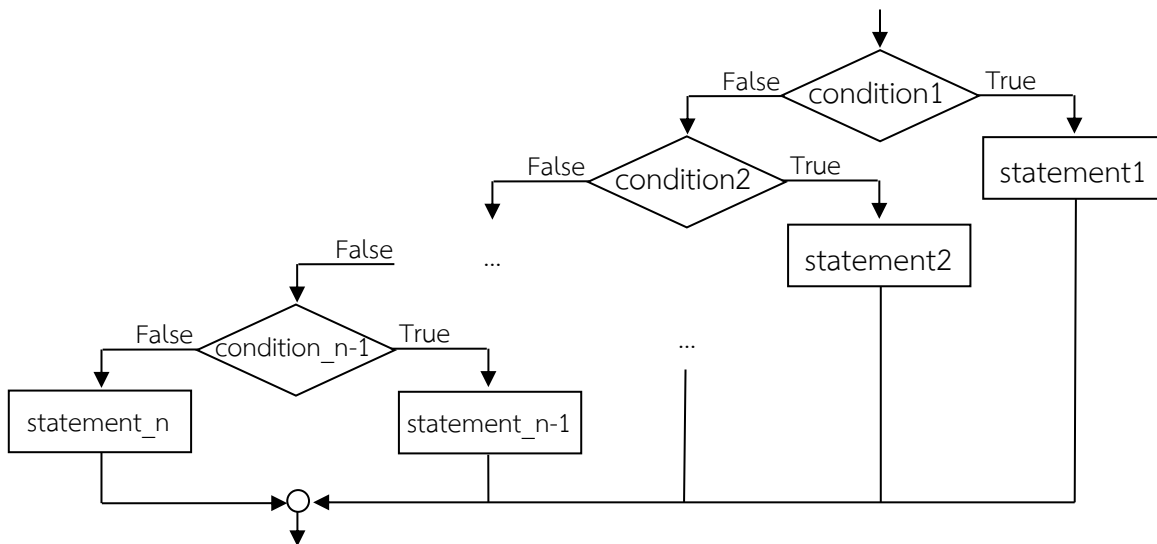
หรือ



คำอธิบาย condition1 , condition2 , ... , condition\_n คือ เงื่อนไขที่กำหนดขึ้น เพื่อใช้ตัดสินว่าจะทำงานหรือไม่ทำงานตามคำสั่ง จะต้องเขียนอยู่หลังคำสั่ง if หรือ elif ซึ่งเงื่อนไขอาจอยู่ในรูปของนิพจน์ (Expression) การคำนวณเปรียบเทียบ หรือเป็นคำสั่งของตัวแปรก็ได้ ที่จะมีการเปรียบเทียบเงื่อนไขว่าเป็นจริง (True) หรือเป็นเท็จ (False)

- ถ้าเงื่อนไขที่ 1 เป็นจริง (True) จะทำ statement1 แล้วออกหรือจบจากคำสั่ง if
- แต่ถ้าเงื่อนไขที่ 1 เป็นเท็จ (False) จะตรวจสอบเงื่อนไขที่ 2 ซึ่งได้ผล 2 แบบ คือ
  1. ถ้าเงื่อนไขที่ 2 เป็นจริง (True) จะทำ statement2 แล้วออกหรือจบจากคำสั่ง if
  - ..... ทดสอบเงื่อนไขต่อไปจนถึง statement\_n-1
  2. ถ้าทุกเงื่อนไขเป็นเท็จ (False) ทั้งหมด จะทำงานหลังคำสั่ง else คือ คำสั่ง statement\_n

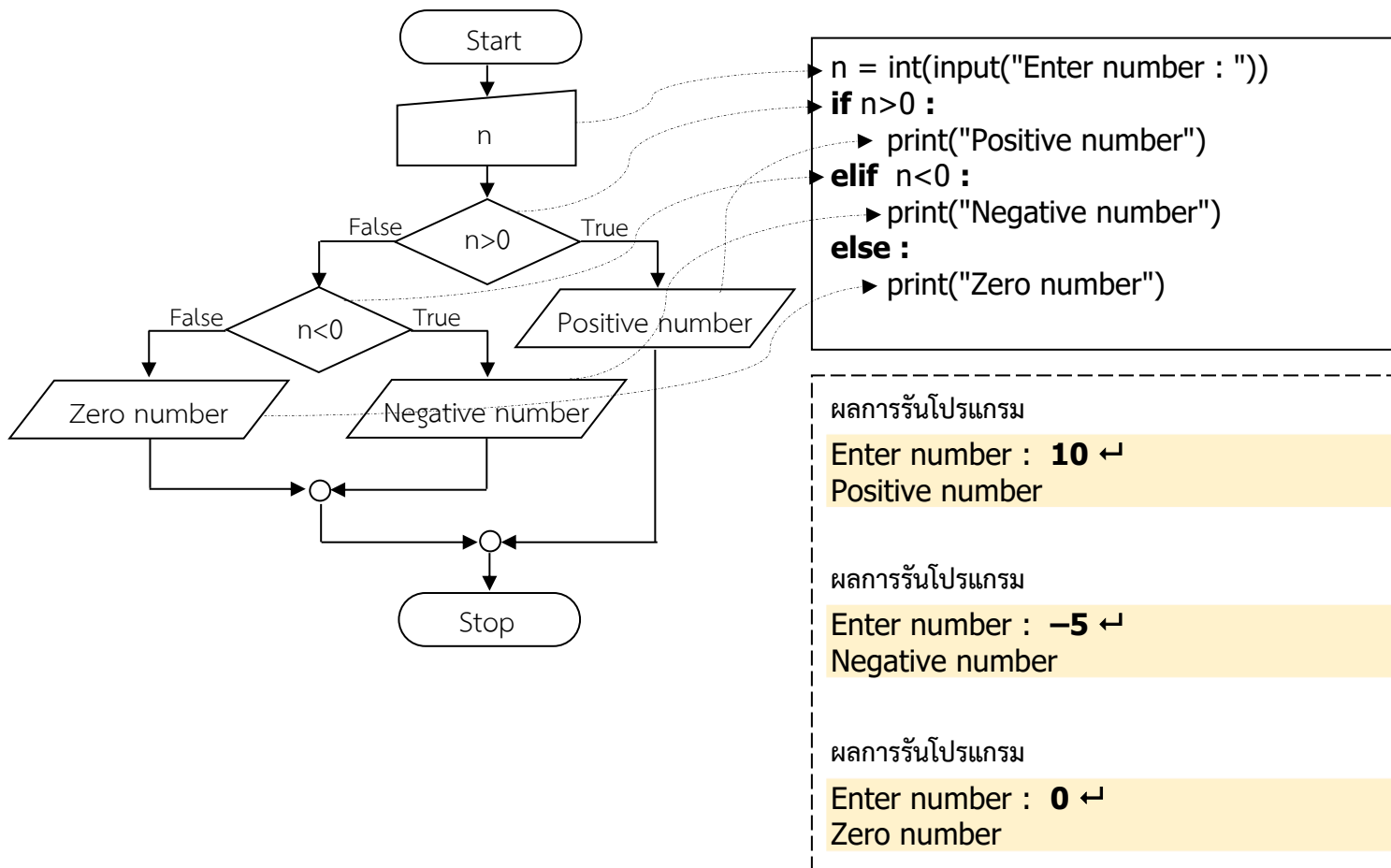




ตัวอย่างโปรแกรมคำสั่งการเลือกทำแบบหลายเส้นทาง (if-elif-else)

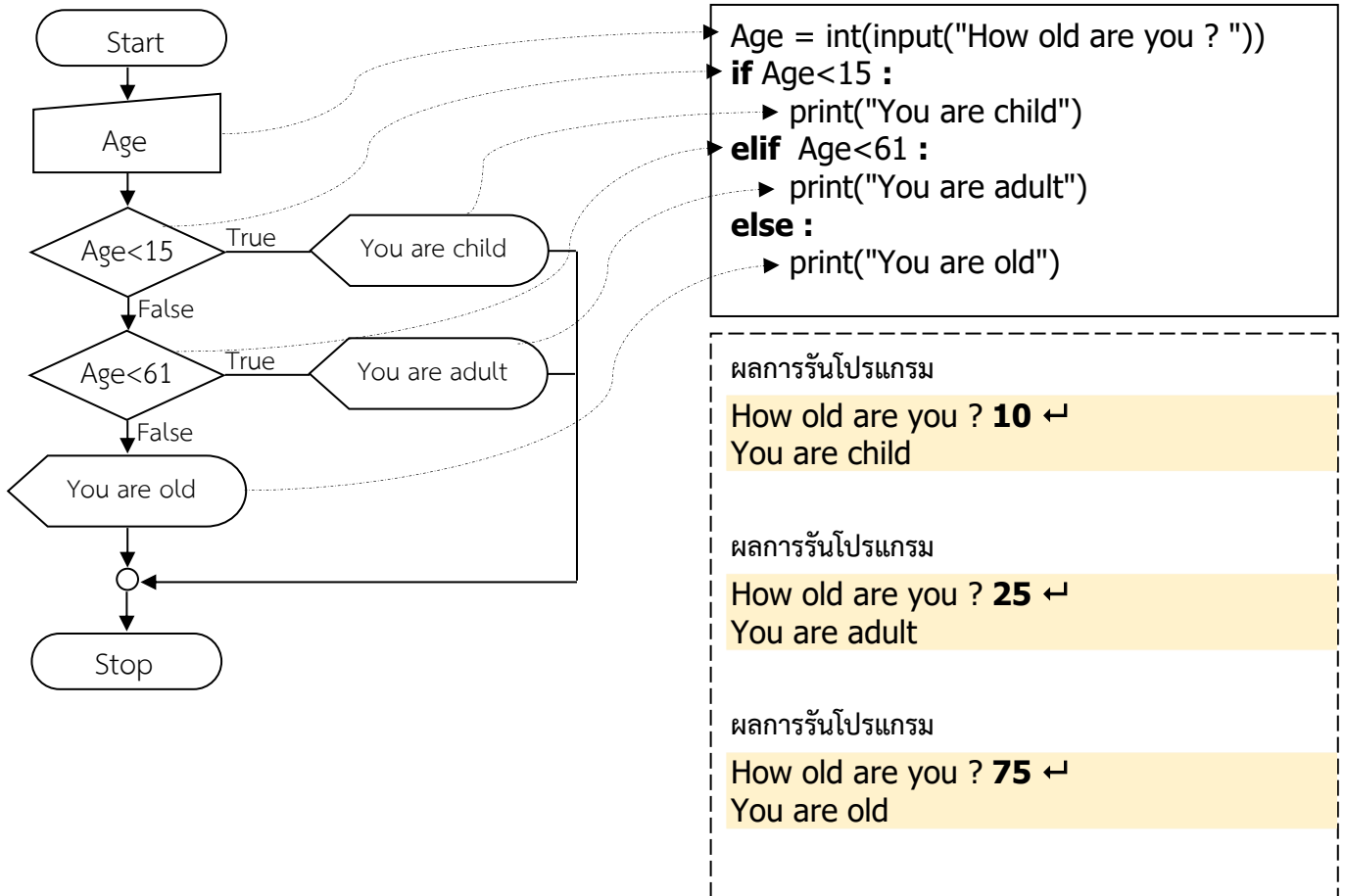
Ex5. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบค่า มีเงื่อนไขการแสดงผลดังนี้

จำนวนเต็ม	แสดงผล
มากกว่า 0	Positive number
น้อยกว่า 0	Negative number
เท่ากับ 0	Zero number



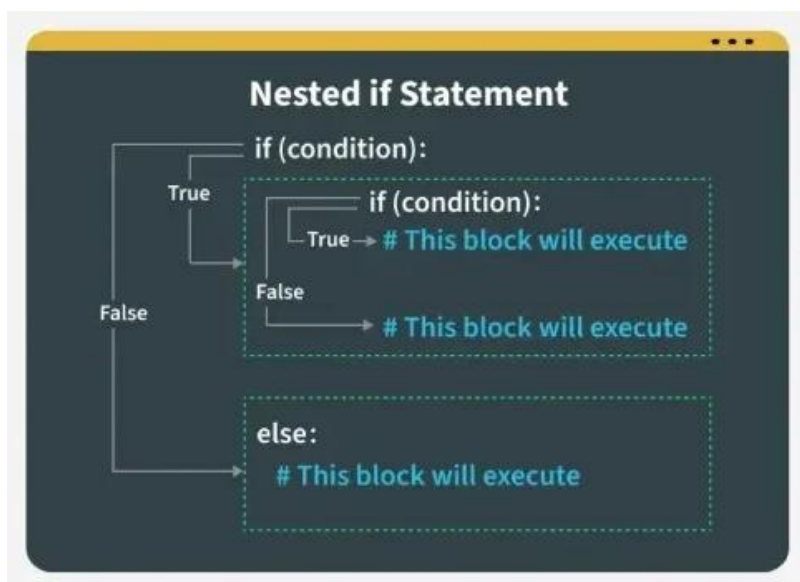
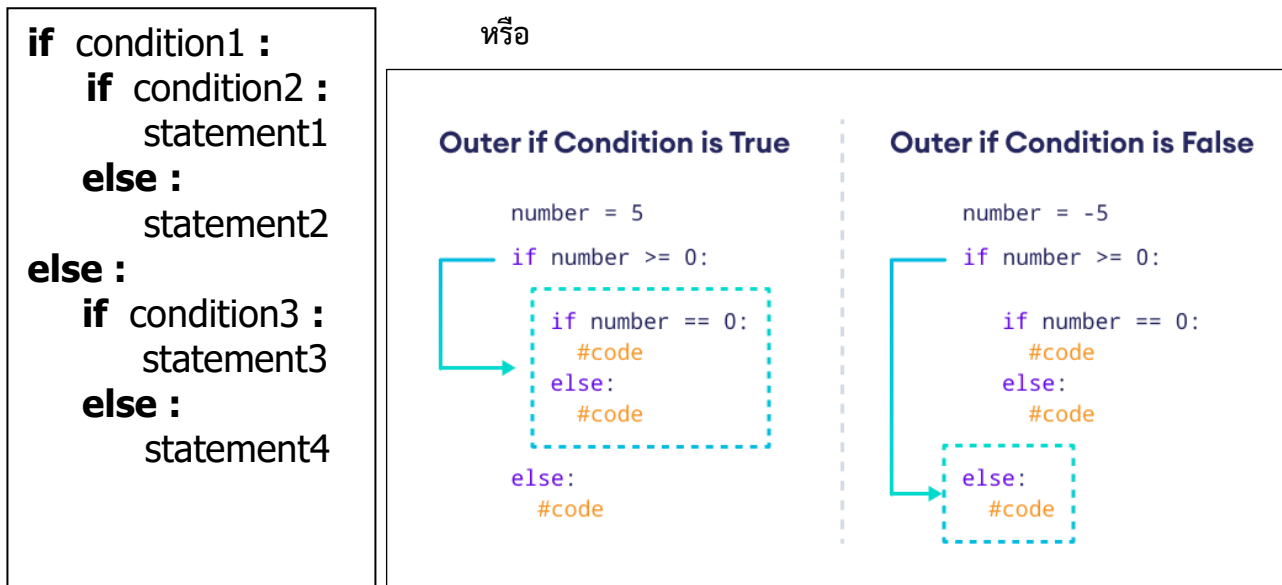
Ex6. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มที่เป็นอายุ แล้วเปรียบเทียบอายุ มีเงื่อนไขการแสดงผลดังนี้

อายุ	แสดงผล
ต่ำกว่า 15 ปี	You are child
15 ถึง 60	You are adult
61 ปีขึ้นไป	You are old



### 3.2 การเลือกทำแบบหลายเส้นทางแบบ Nested if

รูปแบบคำสั่ง

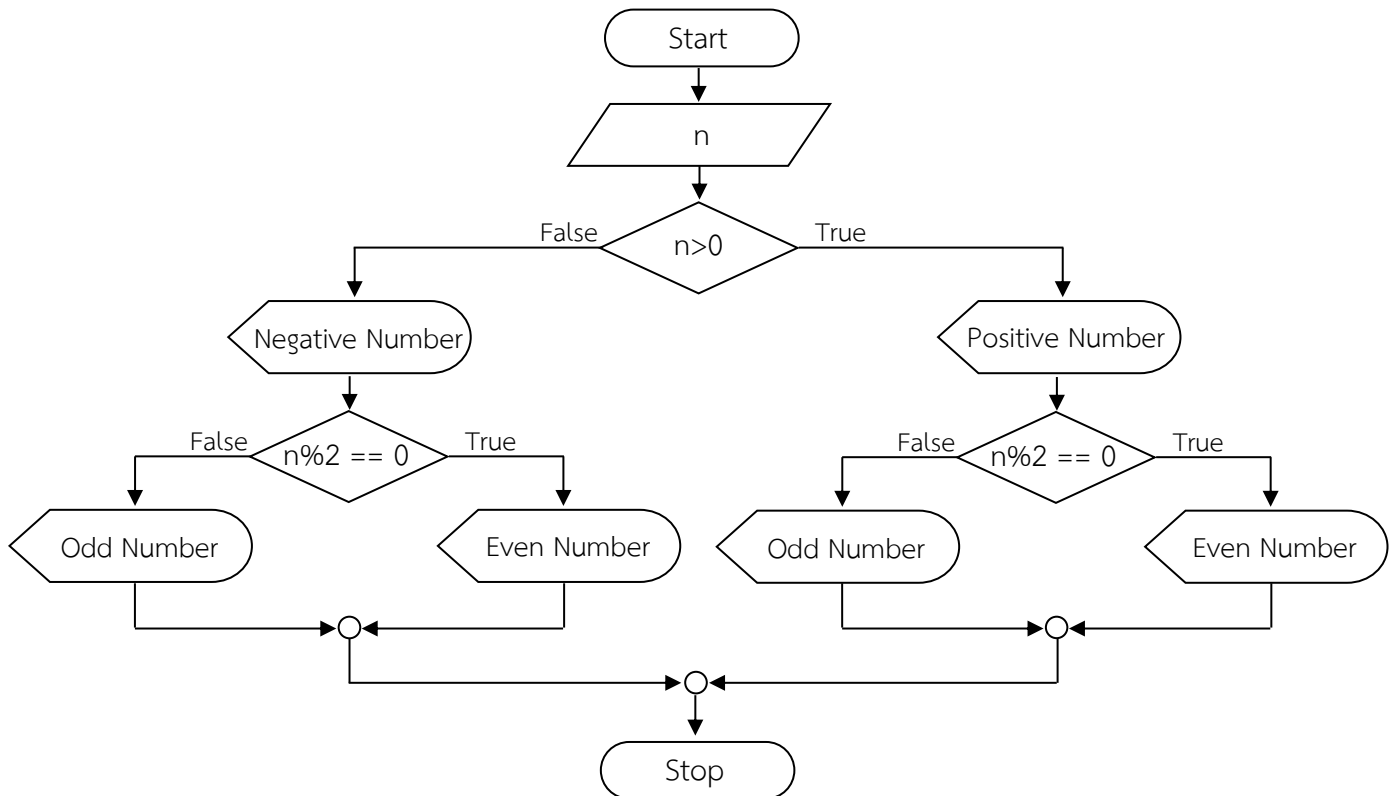


คำอธิบาย condition1 , condition2 , condition\_3 คือ เงื่อนไขที่กำหนดขึ้น เพื่อใช้ตัดสินว่าจะทำงานหรือไม่ทำงานตามคำสั่ง จะต้องเขียนอยู่หลังคำสั่ง if ซึ่งเงื่อนไขอาจอยู่ในรูปของนิพจน์ (Expression) การคำนวณเปรียบเทียบ หรือเป็นคำสั่งของตัวแปรก็ได้ ที่จะมีการเปรียบเทียบเงื่อนไขว่าเป็นจริง (True) หรือเป็นเท็จ (False)

- ถ้าเงื่อนไขที่ 1 (condition1) เป็นจริง (True)
  - ✧ จะทำการเช็คเงื่อนไขที่ 2 (condition2) ว่าเป็นจริง (True) หรือ เป็นเท็จ (False)
    1. ถ้าเงื่อนไขที่ 2 เป็นจริง (True) จะทำ statement1 แล้วออกหรือจบจากคำสั่ง if
    2. ถ้าเงื่อนไขที่ 2 เป็นเท็จ (False) จะทำ statement2 แล้วออกหรือจบจากคำสั่ง if
- แต่ถ้าเงื่อนไขที่ 1 (condition1) เป็นเท็จ (False) จะทำหลังคำสั่ง else
  - ✧ จะทำการเช็คเงื่อนไขที่ 3 (condition3) ว่าเป็นจริง (True) หรือ เป็นเท็จ (False)
    1. ถ้าเงื่อนไขที่ 3 เป็นจริง (True) จะทำ statement3 แล้วออกหรือจบจากคำสั่ง if
    2. ถ้าเงื่อนไขที่ 3 เป็นเท็จ (False) จะทำ statement4 แล้วออกหรือจบจากคำสั่ง if

### ตัวอย่างโปรแกรมคำสั่งการเลือกทำแบบหลายเส้นทาง (Nested if)

Ex7. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบหาค่า ถ้าตัวเลขมากกว่าศูนย์ แล้วแสดงผลเป็นจำนวนเต็มบวก (Positive Number) และแสดงผลว่าเป็นเลขคู่ (Even number) หรือเลขคี่ (Odd number) มิฉะนั้นแสดงผลเป็นจำนวนเต็มลบ (Negative Number) และแสดงผลว่าเป็นเลขคู่ (Even number) หรือเลขคี่ (Odd number)



```

n = int(input("Enter number : "))
if n > 0 :
    print("Positive number")
    if n % 2 == 0 :
        print("Even number")
    else :
        print("Odd number")
else :
    print("Negative number")
    if n % 2 == 0 :
        print("Even number")
    else :
        print("Odd number")
  
```

ผลการรันโปรแกรม (ถ้าเงื่อนไขเป็นจริง)

```

Enter number : 10 ←
Positive number
Even number
  
```

ผลการรันโปรแกรม (ถ้าเงื่อนไขเป็นเท็จ)

```

Enter number : -5 ←
Negative number
Odd number
  
```

### 3.3 การเลือกทำแบบหลายเส้นทางแบบ Match Case

#### รูปแบบคำสั่ง

```

match variable/expression:
  case pattern1 :
    # code block for pattern1
  case pattern2 :
    # code block for pattern2
    ...
  case patternN :
    # code block for patternN
  case _ :
    # Default case (wildcard) if no other pattern matches

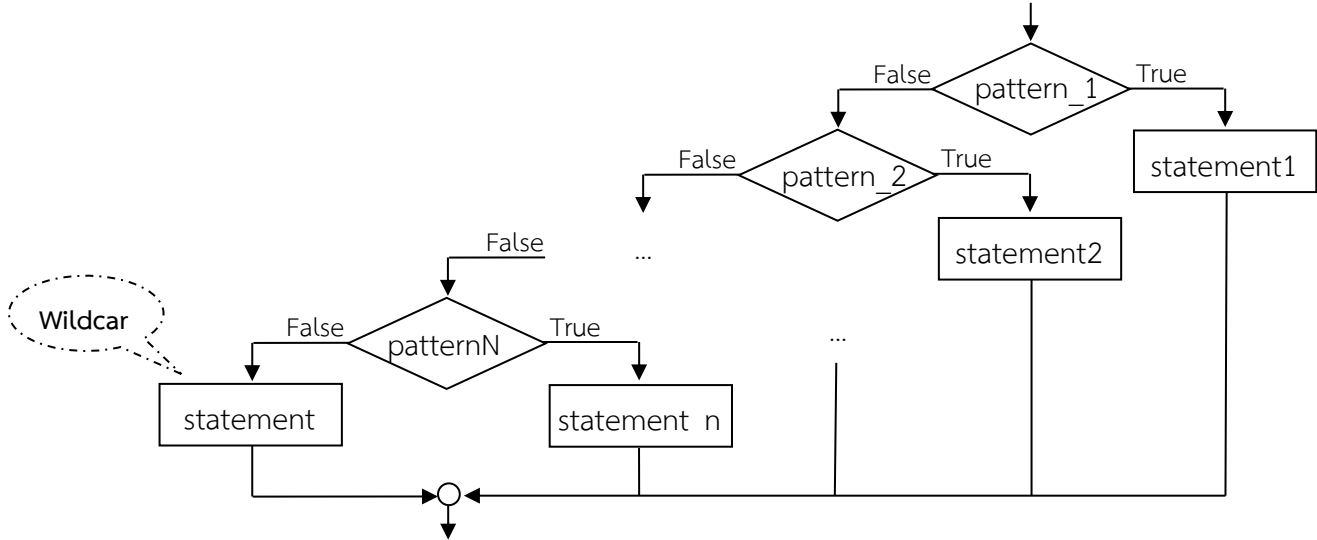
```

คำอธิบาย variable/expression คือ ตัวแปร หรือ นิพจน์ที่ให้ผลออกมาเป็นตัวเลขจำนวนเต็ม หรืออักขระก็ได้ โดยตัวแปรนี้จะถูกใช้เป็นเงื่อนไขในการทำงานต่อไป

pattern\_1, pattern\_2, ... , patternN คือ ค่าคงที่ชนิดจำนวนเต็ม หรือตัวอักขระ โดยต้องเป็นชนิดเดียวกับค่าของตัวแปร (variable) หรือนิพจน์ (expression) เช่น จำนวนเต็ม( 1, 2, 3, 5, 10) อักขระ ( 'a', 'b', 'c', 'd', 'e')

➤ ดังนั้น ถ้าค่าของตัวแปร(variable) หรือนิพจน์ (expression) ตรงกับค่าคงที่ใน pattern ของ case ใด โปรแกรมจะทำงานตามคำสั่งของ case นั้น ๆ

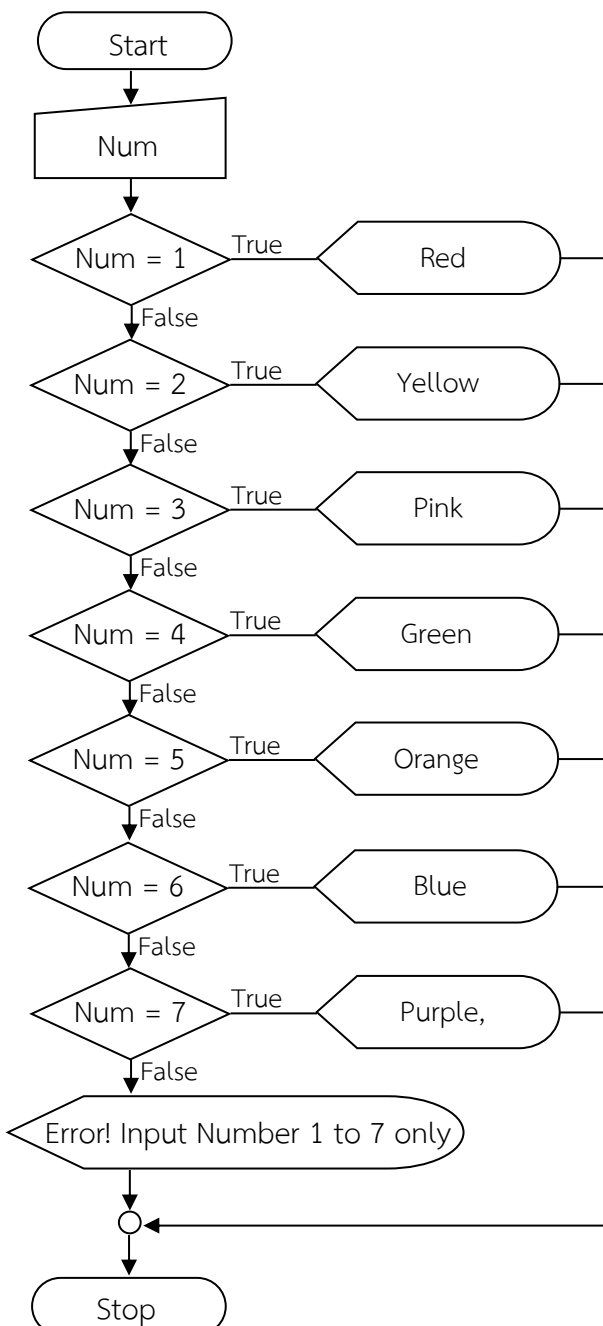
underscore ( \_ ) หรือ Wildcard คือ ถ้าค่าของตัวแปร (variable) หรือนิพจน์ (expression) ไม่ตรงกับค่าของ case ใด ๆ เลย อย่างน้อยโปรแกรมก็ต้องเข้ามาทำงานตามคำสั่งของ default ก่อนออกจากคำสั่ง



## ตัวอย่างโปรแกรมคำสั่งการเลือกทำแบบหลายเส้นทาง (match-case)

Ex7. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบค่า มีเงื่อนไขการแสดงผลดังนี้

จำนวนเต็ม	แสดงผลสีประจำวัน
1	Red
2	Yellow
3	Pink
4	Green
5	Orange
6	Blue
7	Purple, Violet
ตัวเลขอื่น ๆ	Error! Input Number 1 to 7 only



```
Num = int(input("Input Number (1 - 7) : "))
```

```
match Num :
```

```
case 1 : print("Red")
```

```
case 2 : print("Yellow")
```

```
case 3 : print("Pink")
```

```
case 4 : print("Green")
```

```
case 5 : print("Orange")
```

```
case 6 : print("Blue")
```

```
case 7 : print("Purple, Violet")
```

```
case _ : print("Error! Input Number 1 to 7 only")
```

ผลการรันโปรแกรม

```
Input Number (1 - 7) : 3 ↵
Pink
```

ผลการรันโปรแกรม

```
Input Number (1 - 7) : 5 ↵
Orange
```

ผลการรันโปรแกรม

```
Input Number (1 - 7) : 8 ↵
Error! Input Number 1 to 7 only
```

Ex8. จงเขียนโปรแกรมรับค่าเกรดด้วยตัวอักษร 1 ตัวแล้วเปรียบเทียบค่า มีเงื่อนไขการแสดงผลดังนี้

เกรด	แสดงผล
A หรือ a	Excellent
B หรือ b	Good
C หรือ c	Average
D หรือ d	Below Average
E หรือ e	Poor
F หรือ f	Fail
ไม่ตรงกับค่าใด ๆ	Incorrect input

```

grade = input("Enter grade : ")
match grade :
    case "A" | "a" :
        print("Excellent")
    case "B" | "b" :
        print("Good")
    case "C" | "c" :
        print("Average")
    case "D" | "d" :
        print("Below Average")
    case "E" | "e" :
        print("Poor")
    case "F" | "f" :
        print("Grade F")
    case _ :
        print("Incorrect input")

```

ผลการรันโปรแกรม

```

Enter grade : E ←
Poor

```

ผลการรันโปรแกรม

```

Enter grade : b ←
Good

```

ผลการรันโปรแกรม

```

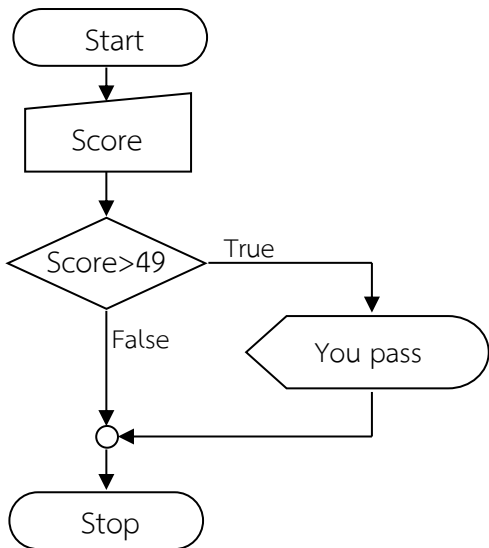
Enter grade : X ←
Incorrect input

```

## ใบงานหน่วยที่ 6

จงเติมคำสั่งที่หายไปให้เป็นคำสั่งที่สมบูรณ์

1. จงเขียนโปรแกรมรับค่าตัวเลขคะแนนรวมของนักเรียน ทำการเปรียบเทียบคะแนน ถ้าคะแนนรวมมากกว่า 49 คะแนน แล้วแสดงผลข้อความสอบผ่าน (You pass)



```
#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if ( )
        printf(" ");
    return 0;
}
```

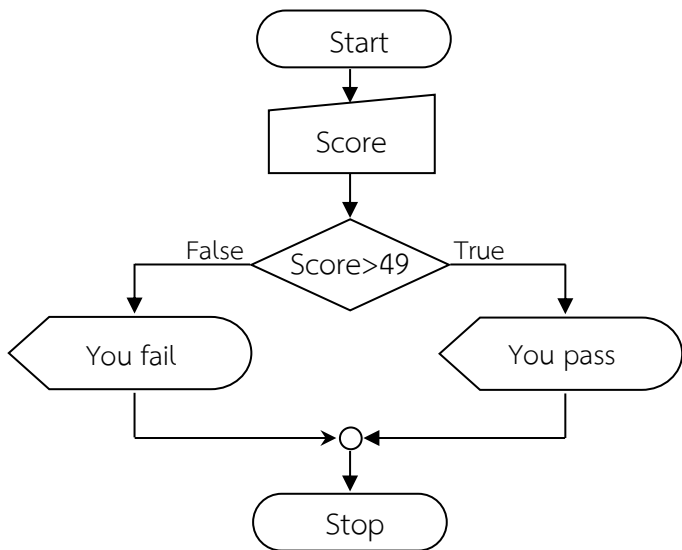
ผลลัพธ์ (ถ้าเงื่อนไขเป็นจริง)

Enter your score : **60**

ผลลัพธ์ (ถ้าเงื่อนไขเป็นเท็จ)

Enter your score : **35**

2. จงเขียนโปรแกรมรับค่าตัวเลขคะแนนรวมของนักเรียน ทำการเปรียบเทียบคะแนน ถ้าคะแนนรวมมากกว่า 49 คะแนน แล้วแสดงผลข้อความสอบผ่าน (You pass) มิฉะนั้นแสดงข้อความสอบตก (You fail)



```
#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if ( )
        printf(" ");
    else
        printf(" ");
    return 0;
}
```

ผลลัพธ์ (ถ้าเงื่อนไขเป็นจริง)

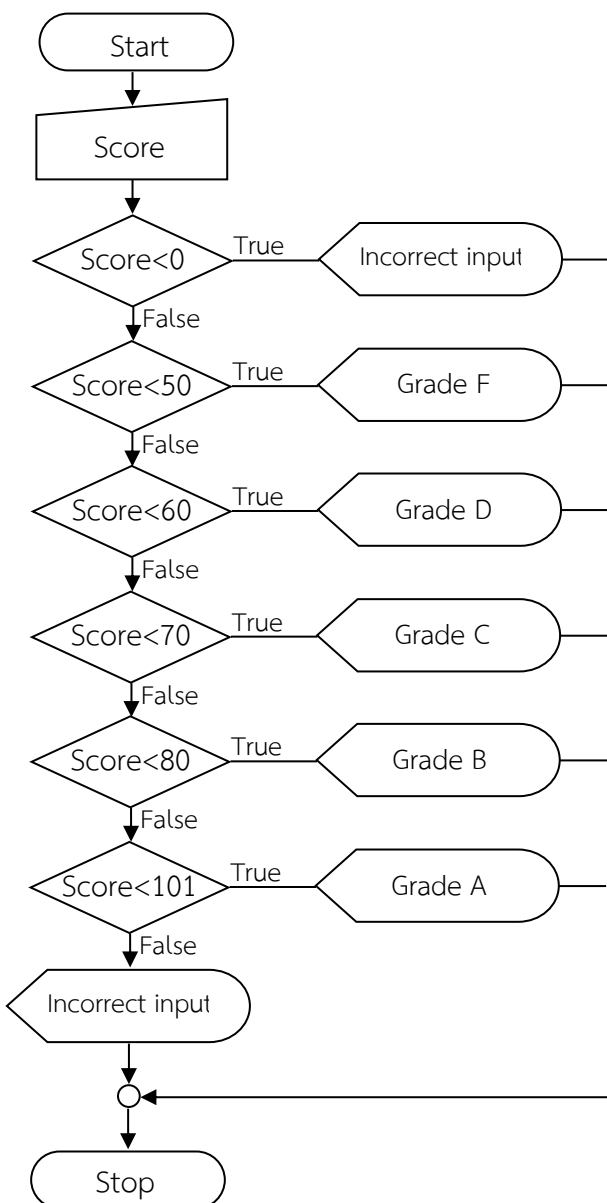
Enter your score : **60**

ผลลัพธ์ (ถ้าเงื่อนไขเป็นเท็จ)

Enter your score : **35**

3. จงเขียนโปรแกรมรับค่าตัวเลขคะแนนรวมของนักเรียน แล้วให้ทำการตัดเกรดจากคะแนนรวมของนักเรียน โดยกำหนดเงื่อนไขดังนี้

คะแนน	แสดงผล
ต่ำกว่า 0	Incorrect input
0 ถึง 49	Grade F
50 ถึง 59	Grade D
60 ถึง 69	Grade C
70 ถึง 79	Grade B
80 ถึง 100	Grade A
101 ขึ้นไป	Incorrect input



```

#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if (Score < 0) printf("Incorrect input ");
    else if (Score < 50) printf("Grade F ");
    else if (Score < 60) printf("Grade D ");
    else if (Score < 70) printf("Grade C ");
    else if (Score < 80) printf("Grade B ");
    else if (Score < 101) printf("Grade A ");
    else printf("Incorrect input ");

    return 0;
}
  
```

ผลลัพธ์

Enter your score : **85**

ผลลัพธ์

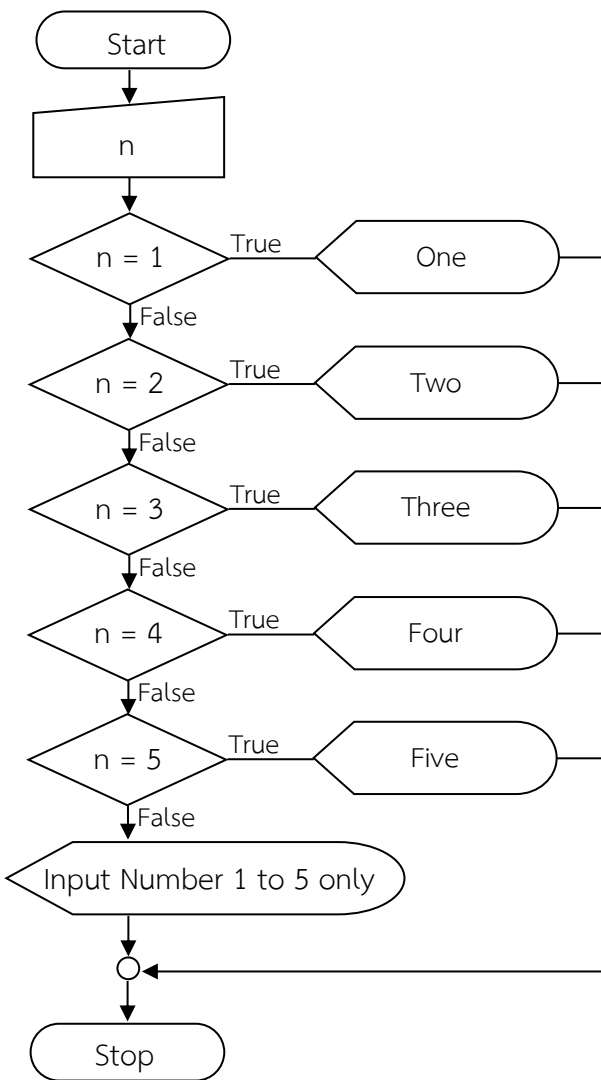
Enter your score : **42**

ผลลัพธ์

Enter your score : **155**

4. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบค่า มีเงื่อนไขการแสดงผลดังนี้

จำนวนเต็ม	แสดงผล
1	One
2	Two
3	Three
4	Four
5	Five
ตัวเลขอื่น ๆ	Input Number 1 to 5 only



```

#include <stdio.h>
int main() {
    int n;
    printf("Input Number (1 - 5) : ");
    scanf("%d" , &n);
    switch (n)
    {
        case 1 : printf("One");
                break;
        case 2 : printf(" ");
                break;
        case 3 : printf("Three");
                break;
        case 4 : printf("Four");
                break;
        case 5 : printf(" ");
                break;
        default : printf("Input Number 1 to 5 only");
    }
    return 0;
}
  
```

ผลลัพธ์

Input Number (1 - 5) : **4**  
 \_\_\_\_\_

ผลลัพธ์

Input Number (1 - 5) : **8**  
 \_\_\_\_\_

ผลลัพธ์

Input Number (1 - 5) : **2**  
 \_\_\_\_\_

5. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มหนึ่งจำนวน แล้วแสดงผลพัธ์ข้อความเป็นเลขคู่ (Even Number) หรือเลขคี่ (Odd Number)

Input	Output
20	Even number
15	Odd number
26	Even number

6. จงเขียนโปรแกรมรับค่าจำนวนเต็มสองจำนวนแล้วให้แสดงผลว่าเท่ากัน (Equal) หรือไม่

Input	Output
20 15	Not equal
12 12	Equal
38 45	Not equal

7. จงเขียนโปรแกรมรับค่าจำนวนเต็มสองจำนวนแล้วให้แสดงผลค่าที่มากที่สุด (Ternary Operator)

Input	Output
20 15	20
12 18	18
38 -5	38

8. จงเขียนโปรแกรมรับค่าจำนวนเต็มสองจำนวน แล้วให้แสดงผลช่วงห่างระหว่างสองค่านั้น

Input	Output
15 35	10
32 14	18
38 -5	43
-15 2	17
-5 -15	10

9. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเงินที่เป็นจำนวนเต็ม 1 จำนวน ให้คำนวณจากตัวเลขจำนวนเต็มที่ได้รับมาว่า จะได้รับเงินธนบัตรใบละ 1000 บาท จำนวนกี่ใบ ธนบัตรใบละ 500 บาท จำนวนกี่ใบ และธนบัตรใบละ 100 บาท จำนวนกี่ใบ โดยมีเงื่อนไขดังนี้

เงื่อนไข	ผลลัพธ์ที่ต้องการ
กรอกตัวเลขเกิน 20,000 บาท	You must enter between 100 – 20000 Baht.
กรอกตัวเลขหลักหน่วยและหลักสิบต่ำกว่า 100 บาท	You must enter for 1000 or 500 or 100 Baht.
กรอกตัวเลขถูกต้อง	แสดงจำนวนธนบัตรใบละ 1000 บาท 500 บาท และ 100 บาท

Input	Output
35600	You must enter between 100 – 20000 Baht.
15600	15 1 1
800	0 1 3
1270	You must enter for 1000 or 500 or 100 Baht.
12400	12 0 4
5823	You must enter for 1000 or 500 or 100 Baht.

10. จงเขียนโปรแกรมรับค่าจำนวนเต็มสามจำนวนแล้วให้แสดงผลค่าน้อยที่สุด (กรอกตัวเลขไม่ซ้ำกัน)

Input	Output
20 15 55	15
12 18 10	12
38 -5 27	-5

11. จงเขียนโปรแกรมรับค่าจำนวนเต็มสามจำนวนแล้วให้แสดงผลค่าที่มากที่สุด (กรอกตัวเลขไม่ซ้ำกัน)

Input	Output
20 15 55	55
12 18 10	18
38 -5 27	38

12. จงเขียนโปรแกรมรับค่าจำนวนเต็มสามจำนวนแล้วให้แสดงผลค่าน้อยที่สุดไปหาค่าที่มากที่สุด (กรอกตัวเลขไม่ซ้ำกัน)

Input	Output
20 15 55	15 20 55
12 18 10	10 12 18
38 -5 27	-5 27 38

13. จงเขียนโปรแกรมรับค่าจำนวนเต็มสามจำนวนแล้วให้แสดงผลค่าที่มากที่สุดไปหาค่าน้อยที่สุด (กรอกตัวเลขไม่ซ้ำกัน)

Input	Output
20 15 55	55 20 15
12 18 10	18 12 10
38 -5 27	38 27 -5

14. การจัดถูงยังชีพหนึ่งถูง ประกอบไปด้วย น้ำ 3 ขวด ขนมปัง 4 ก้อน และไข่ต้ม 2 ฟอง หากกองบรรเทาทุกซีได้รับน้ำ ขนมปังและไข่ต้ม จากผู้บริจาคมาเป็นปริมาณ a ขวด b ก้อน และ c ฟอง ตามลำดับ กองบรรเทาทุกซีจะจัดถูงยังชีพตามข้อกำหนดข้างต้นได้สูงสุดก็ถูง และจะเหลือของบริจาคแต่ละอย่างเป็นปริมาณเท่าใด

จงเขียนโปรแกรมที่รับปริมาณของที่บริจาคเป็นเลขจำนวนเต็ม a, b และ c ตามลำดับ จากนั้นโปรแกรมจะพิมพ์ตัวเลขออกมาสี่จำนวน โดยตัวเลขตัวแรก คือ จำนวนถูงยังชีพที่มากที่สุดที่จัดได้ และตัวเลขสามจำนวนถัดมา คือ ปริมาณน้ำ ขนมปัง และไข่ต้มที่เหลือจากการผลิตตามลำดับ ให้ตัวเลขแต่ละจำนวนคั่นด้วยช่องว่าง 1 ช่อง

Input	Output
10 20 30	3 1 8 24
50 20 30	5 35 0 20
50 70 30	15 5 10 0
30 40 50	10 0 0 30
50 43 21	10 20 3 1

15. ห้างสรรพสินค้าแห่งหนึ่งต้องการคำนวณค่าเช่าสถานที่จอดรถ โดยมีเงื่อนไขดังนี้ (ใช้วิธีคิดแบบก้าวหน้า)

จำนวนชั่วโมงจอดรถ (ชม.)	ค่าเช่าจอดรถ (บาท/ชม.)
ชั่วโมงแรก	30 บาท
ชั่วโมงที่ 2 – 5	ชั่วโมงละ 20 บาท
ชั่วโมงที่ 6 เป็นต้นไป	ชั่วโมงละ 10 บาท

Input	Output
2	50
7	130
5	110
1	30
8	140

16. จงเขียนโปรแกรมคำนวณหาค่าน้ำประปา โดยรับค่าเลขมิเตอร์ก่อนหน้าและครั้งล่าสุด แล้วคำนวณหาค่าน้ำที่ต้องจ่าย โดยมีเงื่อนไขดังนี้ (ใช้วิธีคิดแบบก้าวหน้า)

จำนวนหน่วยที่ใช้	หน่วยละ (บาท)
น้อยกว่าหรือเท่ากับ 10.00	4
ตั้งแต่ 10.01 ถึง 50.00	5
มากกว่า 50.01 ขึ้นไป	6

Input	Output
100 112	50
152 225	378
120 126	24

17. ค่าไฟตามปริมาณการใช้ไฟฟ้า (หน่วยเป็นกิโลวัตต์ชั่วโมงหรือ kWh) แบบก้าวหน้า โดยอัตราค่าบริการแบ่งตามช่วงดังนี้

การใช้ไฟฟ้า (kWh)	ค่าไฟ (บาท/หน่วย)
การใช้ไฟฟ้าไม่เกิน 50 หน่วย	3
การใช้ไฟฟ้า 51-150 หน่วย	5
การใช้ไฟฟ้า 151-300 หน่วย	7
การใช้ไฟฟ้ามากกว่า 300 หน่วย	10

Input	Output
375	2450
45	135
245	1315
80	300

18. คิดภาษีเงินได้ส่วนบุคคลใช้วิธีคิดแบบก้าวหน้า โดยแบ่งเงินได้ออกเป็นช่วง ๆ ดังนี้

รายได้ (บาท)	ภาษี (%)
รายได้ไม่เกิน 150,000 บาท	-
150,001 - 300,000 บาท	5
300,001 - 500,000 บาท	10
รายได้มากกว่า 500,000 บาท	20

Input	Output
250000	5000
142000	0
430000	20500
1200000	167500

19. การคิดค่าน้ำประปาใช้วิธีคิดแบบก้าวหน้า โดยแบ่งตามปริมาณการใช้น้ำเป็นช่วง ๆ ดังนี้

ปริมาณการใช้น้ำ (ลูกบาศก์เมตร (m <sup>3</sup> ))	ค่าน้ำ (บาท/ลูกบาศก์เมตร)
ปริมาณการใช้น้ำไม่เกิน 5 ลูกบาศก์เมตร	10
ปริมาณการใช้น้ำ 6-10 ลูกบาศก์เมตร	15
ปริมาณการใช้น้ำ 11-20 ลูกบาศก์เมตร	20
ปริมาณการใช้น้ำมากกว่า 20 ลูกบาศก์เมตร	25

Input	Output
4	40
16	245
8	95
35	700

20. ค่าจัดส่งตามน้ำหนักของสินค้า โดยใช้อัตราค่าบริการแบบก้าวหน้า ดังนี้

น้ำหนัก (กิโลกรัม)	ค่าจัดส่ง (บาท)
น้ำหนักไม่เกิน 2 กิโลกรัม	30 บาท
น้ำหนัก 2.01 - 5 กิโลกรัม	50 บาท
น้ำหนัก 5.01 - 10 กิโลกรัม	80 บาท
น้ำหนักมากกว่า 10 กิโลกรัม	100 บาท + 10 บาทต่อกิโลกรัมที่เกินจาก 10 กิโลกรัม

Input	Output
13.5	135.00
2.5	50.00
8.2	80.00
1.7	30.00
22.75	227.50

21. จงเขียนโปรแกรมรับค่าตัวเลขจำนวนเต็มที่เป็นคำตอบ โดยให้ผู้เล่นคำนวณหาผลลัพธ์จากการคูณของตัวเลขทั้งสามจำนวน แล้วให้ระบบสุ่มตัวเลขหนึ่งหลักสามจำนวนมาให้ จาก 1 – 9 (คำสั่งสุ่มค่าใช้ฟังก์ชัน rand())

Input	Output
What is $9 \times 4 \times 7 = \mathbf{252}$	$9 \times 4 \times 7 = 252$ is true.
What is $3 \times 8 \times 5 = \mathbf{129}$	$3 \times 8 \times 5 = 129$ is false.
What is $2 \times 4 \times 5 = \mathbf{40}$	$2 \times 4 \times 5 = 40$ is true.
What is $8 \times 6 \times 9 = \mathbf{432}$	$8 \times 6 \times 9 = 432$ is true.
What is $7 \times 1 \times 3 = \mathbf{32}$	$7 \times 1 \times 3 = 32$ is false.

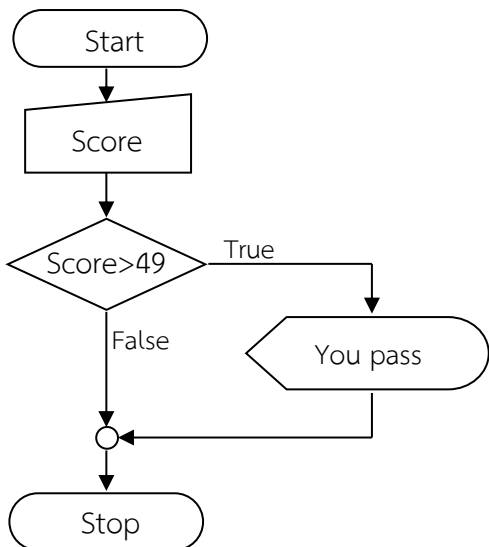
22. จงเขียนโปรแกรมเป่ายิงฉุบ หรือ กรรไกร-ก้อนหิน-กระดาษ (Scissor-Rock-Paper) ระหว่างคอมพิวเตอร์กับผู้เล่น โดยคอมพิวเตอร์จะสุ่มค่าเป็น 0 , 1 หรือ 2 หมายถึง 0-กรรไกร, 1-ก้อนหิน, 2-กระดาษ ตามลำดับ (การสุ่มค่าใช้ฟังก์ชัน rand()) และผู้เล่นจะรับค่าผ่านทางแป้นพิมพ์ เป็น 0 , 1 หรือ 2 จากนั้นจะแสดงผลการเล่นเป็น ชนะ (won) , แพ้ (lost) หรือ เสมอ (tie/draw)

Input	Output
*** Pao-Ying-Choob *** Scissor(0), Rock(1), Paper(2) : <b>1</b>	The computer is scissor. Yor are rock. It is a won.
*** Pao-Ying-Choob *** Scissor(0), Rock(1), Paper(2) : <b>2</b>	The computer is paper. Yor are paper too. It is a tie/draw.
*** Pao-Ying-Choob *** Scissor(0), Rock(1), Paper(2) : <b>0</b>	The computer is rock. Yor are scissor. It is a lost.
*** Pao-Ying-Choob *** Scissor(0), Rock(1), Paper(2) : <b>1</b>	The computer is paper. Yor are rock. It is a lost.

### เฉลยใบงานหน่วยที่ 6

จงเติมคำสั่งที่หายไปให้เป็นคำสั่งที่สมบูรณ์

- รับค่าตัวเลขคะแนนรวมของนักเรียน ทำการเปรียบเทียบคะแนน ถ้าคะแนนรวมมากกว่า 49 คะแนน แล้วแสดงผลลัพธ์ข้อความสอบผ่าน (You pass)

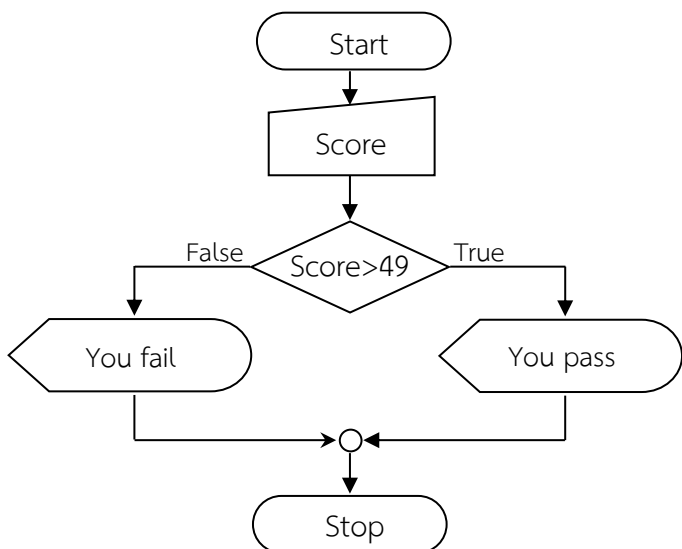


```
#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if (Score>49)
        printf("You Pass");
    return 0;
}
```

ผลลัพธ์ (ถ้าเงื่อนไขเป็นจริง)  
 Enter your score : 60  
 You pass

ผลลัพธ์ (ถ้าเงื่อนไขเป็นเท็จ)  
 Enter your score : 35

- รับค่าตัวเลขคะแนนรวมของนักเรียน ทำการเปรียบเทียบคะแนน ถ้าคะแนนรวมมากกว่า 49 คะแนน แล้วแสดงผลลัพธ์ข้อความสอบผ่าน (You pass) มิฉะนั้นแสดงข้อความสอบตก (You fail)



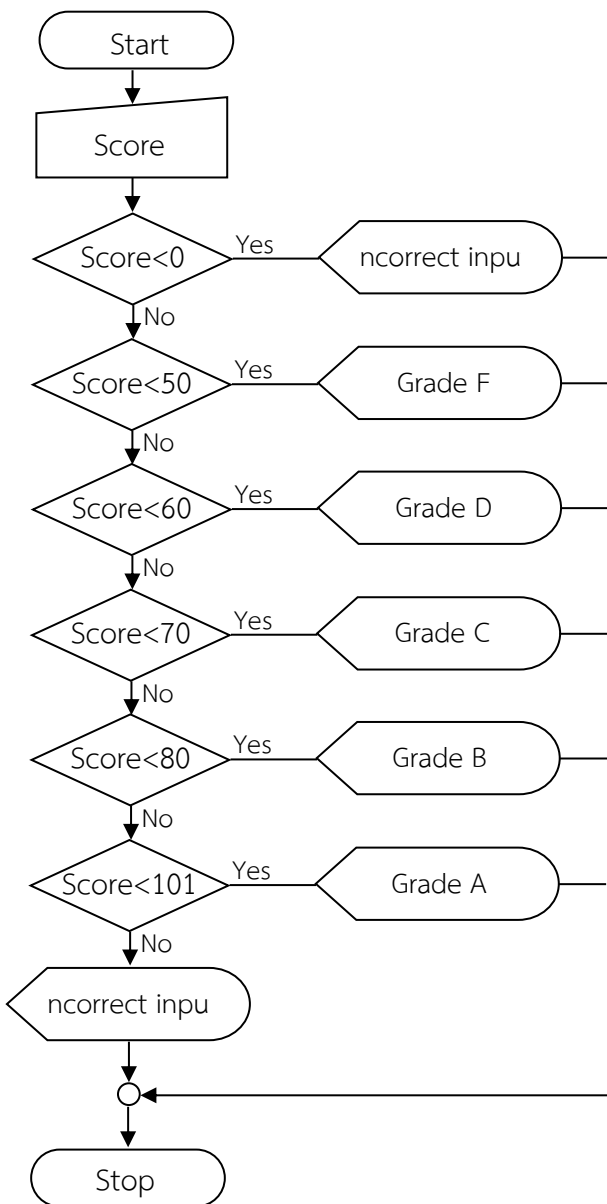
```
#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if (Score>49)
        printf("You pass");
    else
        printf("You false");
    return 0;
}
```

ผลลัพธ์ (ถ้าเงื่อนไขเป็นจริง)  
 Enter your score : 60  
 You pass

ผลลัพธ์ (ถ้าเงื่อนไขเป็นเท็จ)  
 Enter your score : 35

3. รับค่าตัวเลขคะแนนรวมของนักเรียน แล้วให้ทำการตัดเกรดจากคะแนนรวมของนักเรียน กำหนดเงื่อนไขดังนี้

คะแนน	แสดงผล
ต่ำกว่า 0	Incorrect input
0 ถึง 49	Grade F
50 ถึง 59	Grade D
60 ถึง 69	Grade C
70 ถึง 79	Grade B
80 ถึง 100	Grade A
101 ขึ้นไป	Incorrect input



```

#include <stdio.h>
int main() {
    float Score;
    printf("Enter your score : ");
    scanf("%f", &Score);
    if (Score < 0) printf("Incorrect input");
    else if (Score < 50) printf("Grade F");
    else if (Score < 60) printf("Grade D");
    else if (Score < 70) printf("Grade C");
    else if (Score < 80) printf("Grade B");
    else if (Score < 101) printf("Grade A");
    else printf("Incorrect input");

    return 0;
}
  
```

ผลลัพธ์

Enter your score : 85  
Grade A

ผลลัพธ์

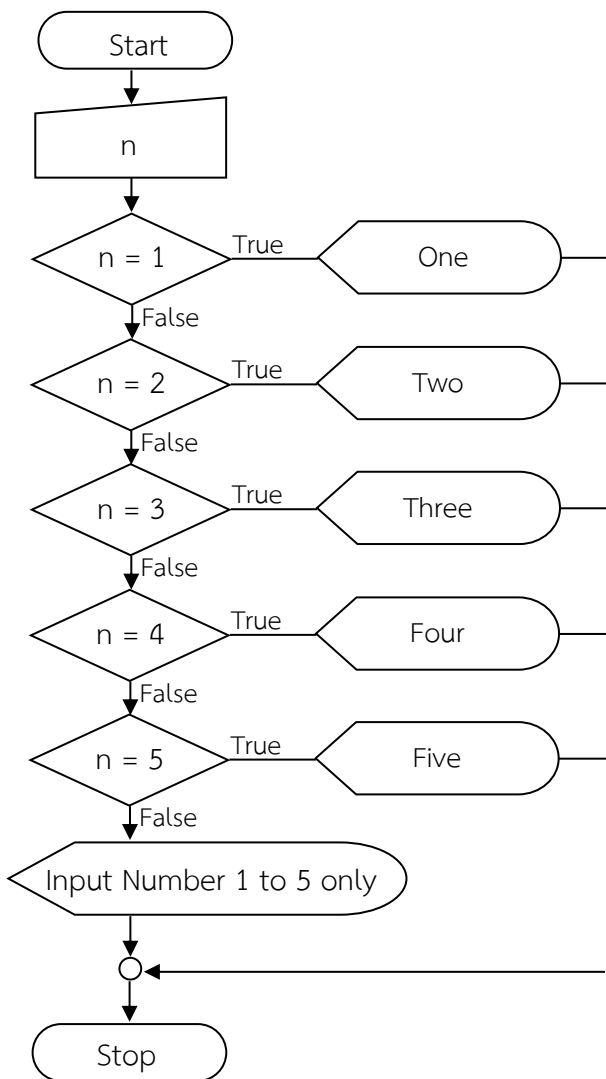
Enter your score : 42  
Grade F

ผลลัพธ์

Enter your score : 155  
Incorrect input

4. รับค่าตัวเลขจำนวนเต็ม 1 จำนวน แล้วเปรียบเทียบค่า มีเงื่อนไขการแสดงผลดังนี้

จำนวนเต็ม	แสดงผล
1	One
2	Two
3	Three
4	Four
5	Five
ตัวเลขอื่น ๆ	Input Number 1 to 5 only



```

#include <stdio.h>
int main() {
    int n;
    printf("Input Number (1 - 5) : ");
    scanf("%d" , &n);
    switch (n)
    {
        case 1 : printf("One");
                break;
        case 2 : printf("Two");
                break;
        case 3 : printf("Three");
                break;
        case 4 : printf("Four");
                break;
        case 5 : printf("Five");
                break;
        default : printf("Input Number 1 to 5 only");
    }
    return 0;
}
    
```

ผลลัพธ์

Input Number (1 - 5) : 4  
Four

ผลลัพธ์

Input Number (1 - 5) : 8  
Input Number 1 to 5 only

ผลลัพธ์

Input Number (1 - 5) : 2  
Two

**Ex8.** รับค่าตัวเลขจำนวนเต็มหนึ่งจำนวน ทำการคำนวณเปรียบเทียบตัวเลขนั้น ถ้าคะแนนรวมมากกว่า 49 คะแนน แล้วแสดงผลพร้อมข้อความจำนวนคู่ (Even Number) มิฉะนั้นแสดงข้อความจำนวนคี่ (Odd Number)

**Ex8.** รับค่าตัวเลขจำนวนเต็มหนึ่งจำนวน ทำการคำนวณเปรียบเทียบตัวเลขนั้น แล้วแสดงผลพร้อมข้อความเป็นเลขคู่ (Even Number) หรือเลขคี่ (Odd Number)

Input	Output
10	Even number
3	Odd number
52	Even number

**Ex9.** ให้เขียนโปรแกรมตรวจสอบว่า ปี พ.ศ. ที่สนใจเป็นปีอธิกสุรทินหรือไม่ ซึ่งปีอธิกสุรทิน (Leap Year) คือ ปี ค.ศ. ที่หารด้วย 4 ลงตัว แต่หารด้วย 100 ไม่ลงตัว หรือหารด้วย 400 ลงตัว

Input	Output
10	Leap Year
3	Leap Year
52	Leap Year