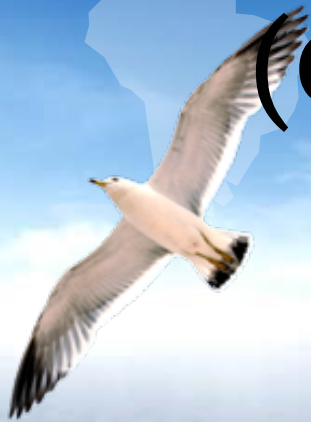


บทที่ 5

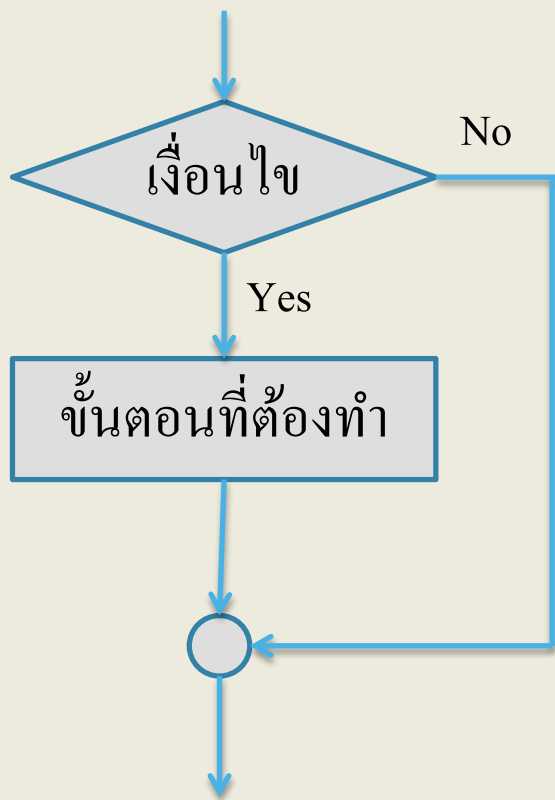
คำสั่งตรวจสอบเงื่อนไข (Condition Statement)



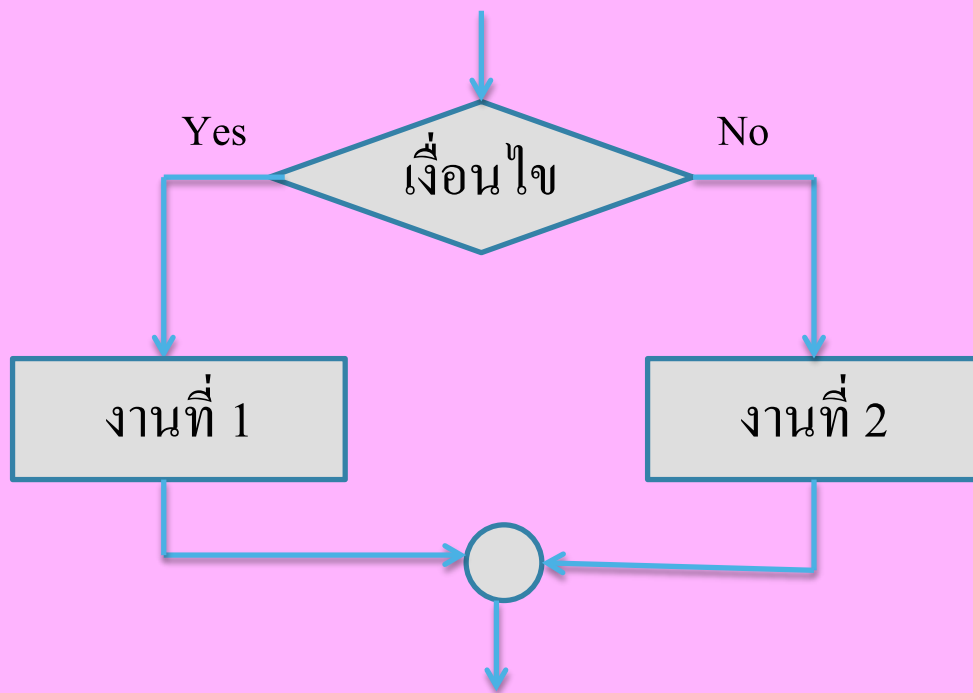


เงื่อนไข

แบบทางเดียว

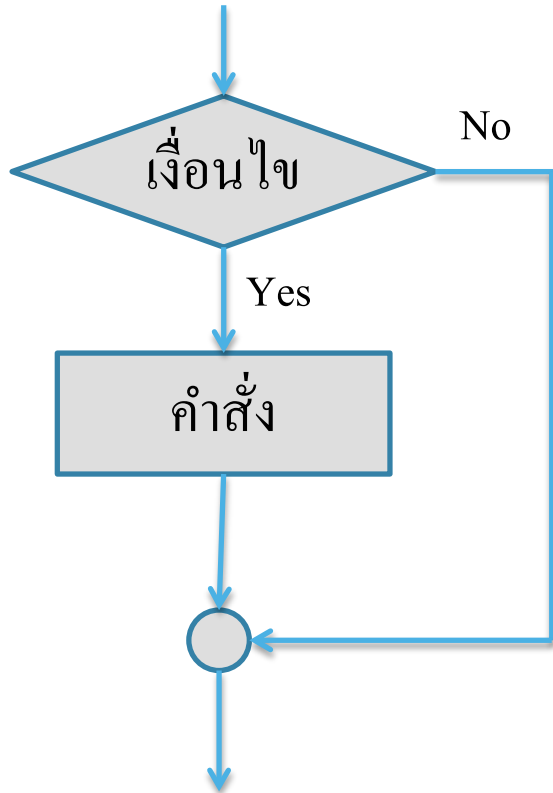


แบบ 2 ทาง





เงื่อนไขทางเดียว



รูปแบบคำสั่งภาษาซี

```
if (เงื่อนไข) {  
    คำสั่ง;  
}
```



เงื่อนไขทางเดียว

```
if (เงื่อนไข) {  
    คำสั่งที่1;  
    คำสั่งที่2;  
}  
คำสั่งที่3;  
คำสั่งที่4;
```

ลำดับการทำงาน

1. ทดสอบนิพจน์เงื่อนไข ซึ่งจะ ได้ผลลัพธ์ เป็นจริง หรือ เท็จ
2. หากนิพจน์เป็นจริง จะทำคำสั่งที่อยู่ ภายใต if ที่อยู่ใน { }
3. หากนิพจน์เป็นเท็จ จะไม่สนใจคำสั่ง ภายใต if จะข้ามไปยังคำสั่งถัดไป

ตัวดำเนินการเปรียบเทียบ

!!!! คำเตือน !!!!



= ใช้กำหนดค่า

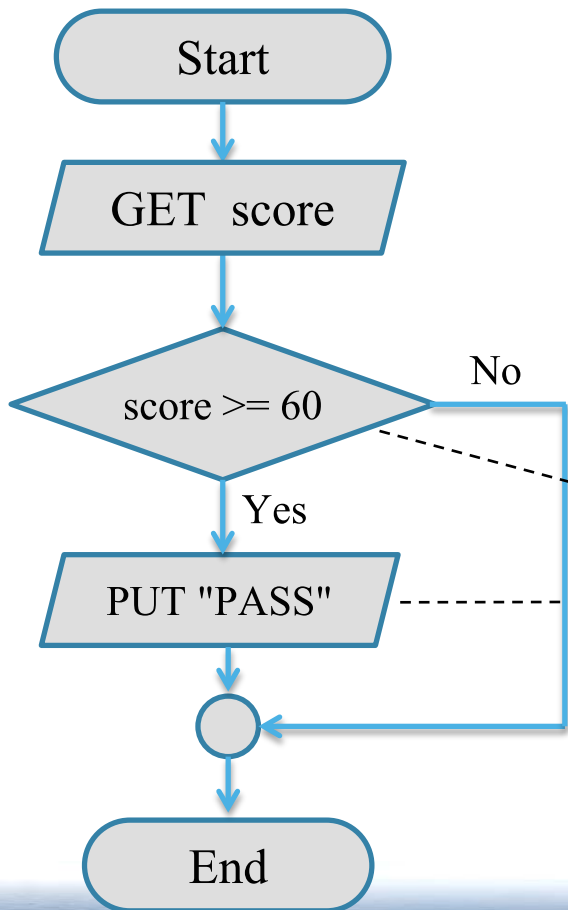
== ใช้เปรียบเทียบค่า

ตัวดำเนินการ	กระบวนการ
==	เท่ากับ
!=	ไม่เท่ากับ
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
>	มากกว่า
<	น้อยกว่า



ตัวอย่าง

โปรแกรมรับค่าคะแนนจากผู้ใช้ หากคะแนนที่ป้อนเข้ามามีค่ามากกว่าหรือเท่ากับ 60 จะพิมพ์ข้อความว่า "PASS" แต่จะไม่พิมพ์ข้อความใดในกรณีอื่น



```
1 #include <stdio.h>
2
3 void main() {
4     int score;
5     printf("Enter score : ");
6     scanf("%d", &score);
7
8     if (score >= 60) {
9         printf("PASS\n");
10    }
11 }
```



ภาษาซี 1 คือ *true* , 0 คือ *false*

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

```
void main() {  
    int a=4, b=4;  
    printf("%d\n", a==b);  
    printf("%d\n", a!=b);  
    printf("%d\n", a>b);  
    printf("%d\n", a>=b);  
    printf("%d\n", a<b);  
    printf("%d\n", a<=b);  
}
```

C:\Users\Theera

```
1  
0  
0  
1  
1  
0  
1  
1
```



ตัวดำเนินการตรรกะ

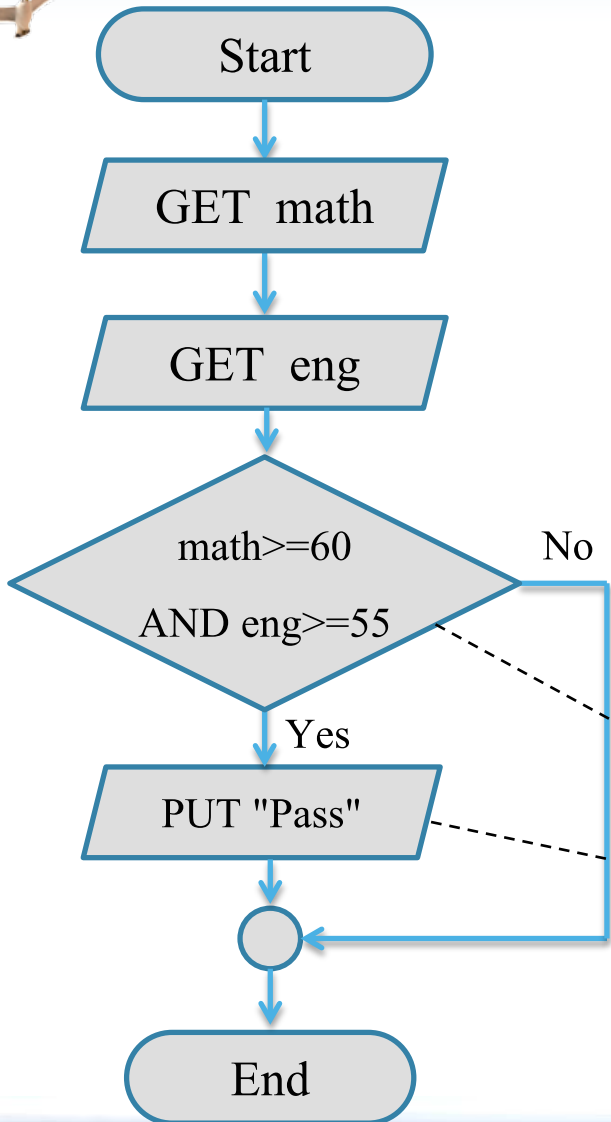
ตัวดำเนินการ	การกระทำ
&&	AND ถ้าค่าทั้งสองเป็นจริงผลลัพธ์จะเป็นจริง
	OR ถ้าทั้งสองเป็นเท็จผลลัพธ์จะเป็นเท็จ
!	NOT เปลี่ยนค่าจากจริงเป็นเท็จ หรือ จากเท็จเป็นจริง

A	B	A && B	A B	!A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True



ตัวอย่าง

โปรแกรมรับค่าคะแนน 2 วิชาจากผู้ใช้ หากคะแนนคณิตศาสตร์มากกว่าเท่ากับ 60 คะแนน และคะแนนภาษาอังกฤษมากกว่าเท่ากับ 55 คะแนน จะพิมพ์ข้อความว่า "PASS" แต่จะไม่พิมพ์ข้อความใดในกรณีอื่น

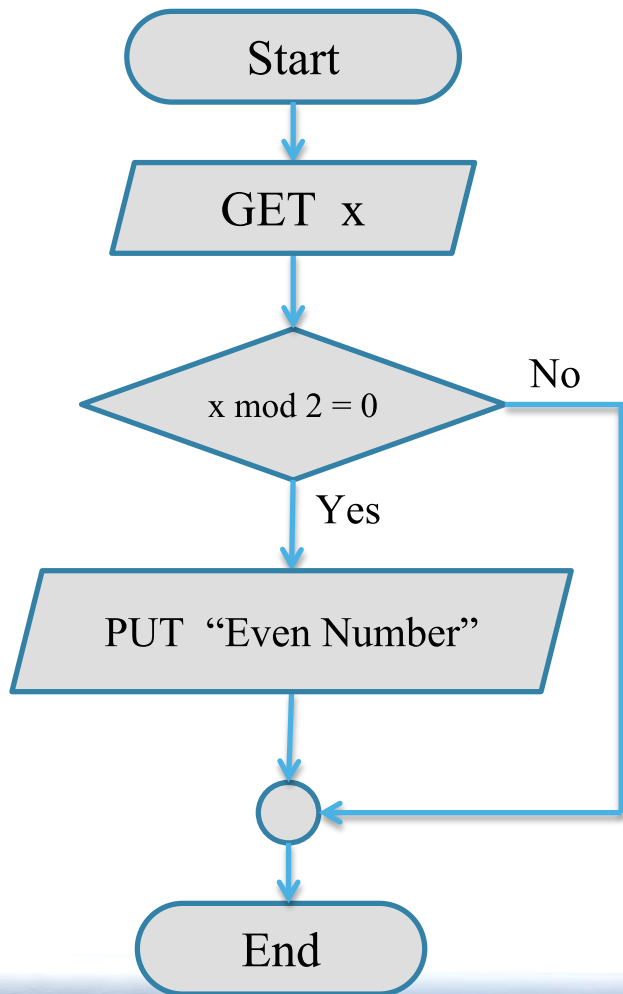


```
1 #include <stdio.h>
2 void main() {
3     int math, eng;
4     printf("Enter math score :");
5     scanf("%d", &math);
6     printf("Enter english score :");
7     scanf("%d", &eng);
8
9     if (math >= 60 && eng >= 55) {
10        printf("Pass\n");
11    }
12 }
```



กิจกรรม

โปรแกรมหาว่าตัวเลขที่รับเข้ามาเป็นเลขคู่ (Even Number) หรือไม่ จงเขียนเขียนโปรแกรมภาษาซีจากขั้นตอนวิธีนี้

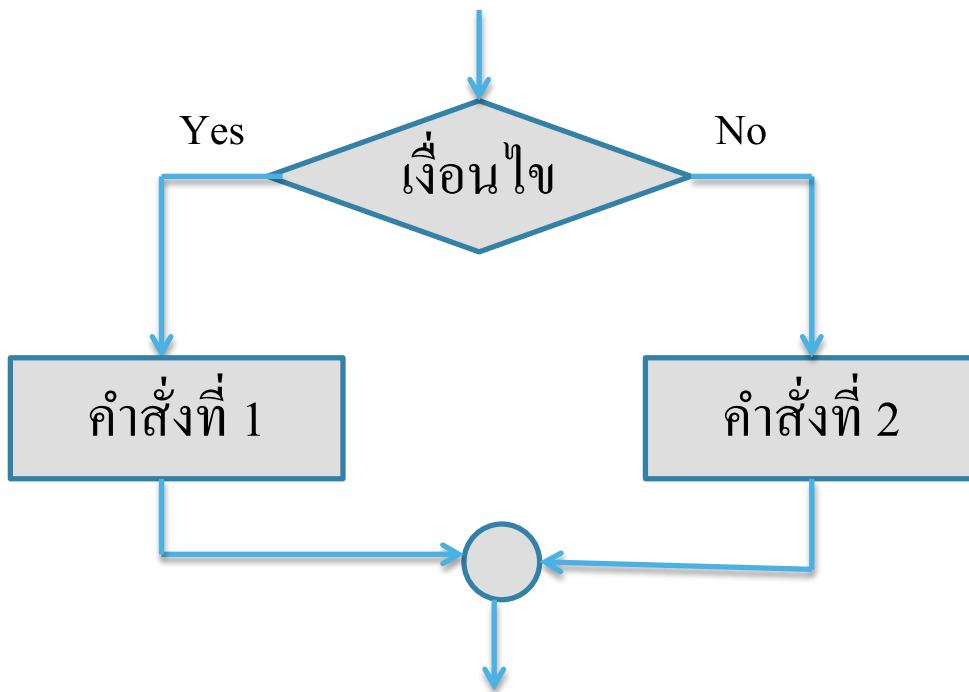


```
#include <stdio.h>
void main() {
    int x;
    printf("Enter a number: ");
    scanf("%d", &x);
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
}
```



เงื่อนไข 2 ทาง

รูปแบบคำสั่งภาษาซี

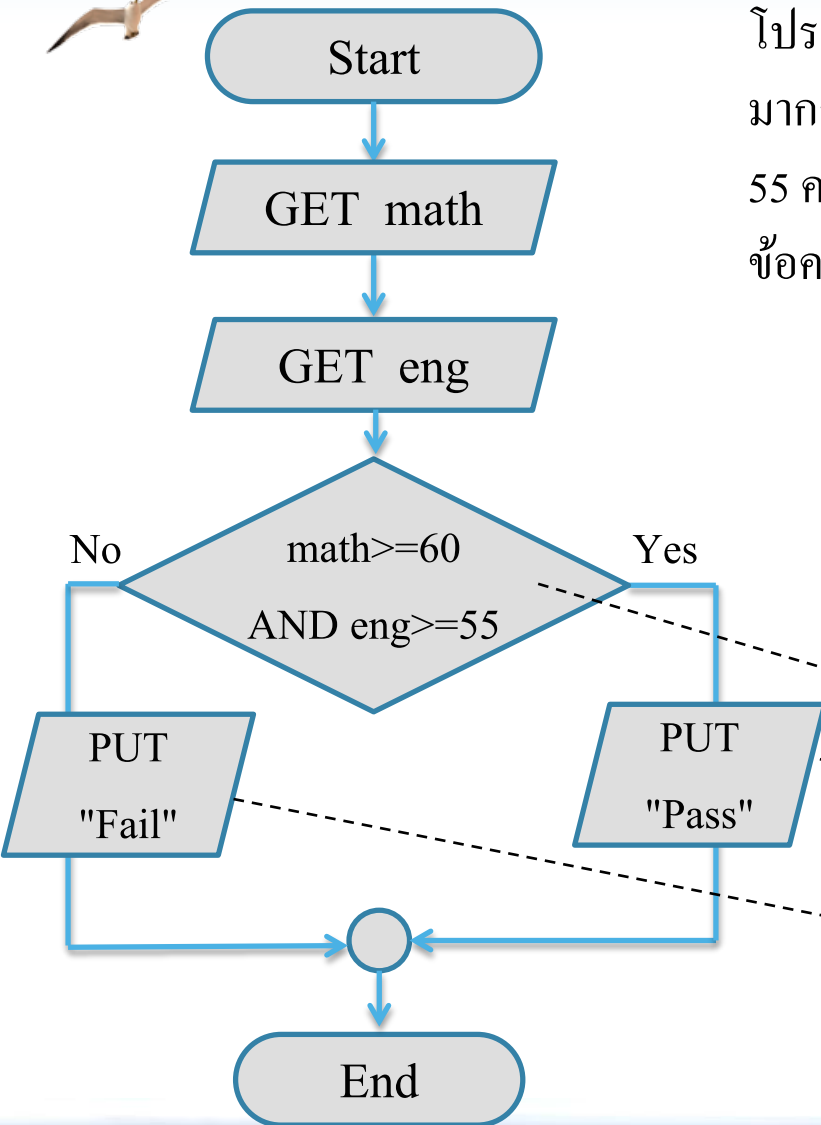


```
if (เงื่อนไข) {  
    คำสั่งที่1;  
} else {  
    คำสั่งที่2;  
}
```



เงื่อนไข 2 ทาง

โปรแกรมรับค่าคะแนน 2 วิชาจากผู้ใช้ หากคะแนนคณิตศาสตร์มากกว่าเท่ากับ 60 คะแนน และคะแนนภาษาอังกฤษมากกว่าเท่ากับ 55 คะแนน จะพิมพ์ข้อความว่า "PASS" แต่หากไม่ใช่จะพิมพ์ข้อความ "Fail"



```
1 #include <stdio.h>
2 void main() {
3     int math, eng;
4     printf("Enter math score :");
5     scanf("%d", &math);
6     printf("Enter english score :");
7     scanf("%d", &eng);
8
9     if (math >= 60 && eng >= 55) {
10         printf("Pass\n");
11     } else {
12         printf("Fail\n");
13     }
14 }
```



กิจกรรม

จงเขียน Algorithm และสร้างโปรแกรมสำหรับตรวจสอบค่าตัวเลข
2 ตัว ว่ามีค่ามากกว่า น้อยกว่า หรือเท่ากับ
ตัวอย่างการทำงานของโปรแกรมหาดังนี้

Test case 1:

Input the first number: 5 ↵
Input the second number: 5↵
Result: 5 = 5

Test case 2:

Input the first number: 3 ↵
Input the second number: 7↵
Result: 3 < 7

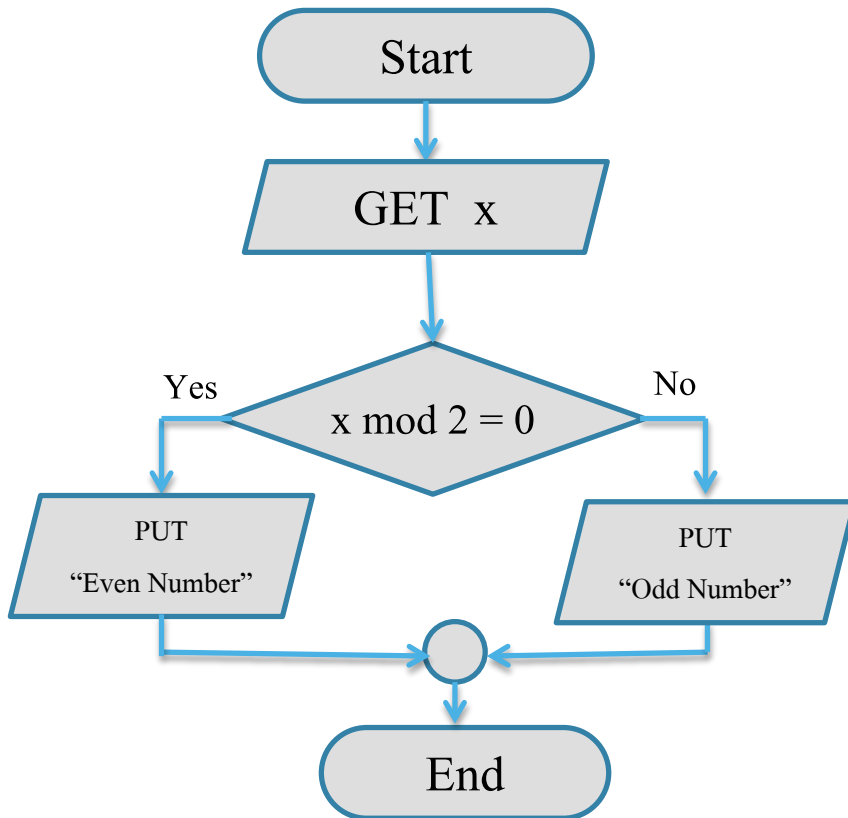
Test case 3:

Input the first number: 7 ↵
Input the second number: 3↵
Result: 7 > 3



กิจกรรม

โปรแกรมหาว่าตัวเลขที่รับเข้ามาเป็นเลขคู่ (Even Number) หรือเลขคี่ (Odd Number)
จงเขียนเขียน โปรแกรมภาษาซีจากขั้นตอนวิธีนี้



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

```
void main() {
```

```
    int x;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &x);
```

```
    .....  
    .....  
    .....  
    .....  
    .....  
    .....  
    .....  
    .....  
    .....
```

```
}
```



การใส่ { } ในคำสั่ง if

กรณีที่มีเพียง 1 คำสั่งภายใน if หรือ else ไม่ต้องใส่ปีกกาครอบก็ได้ แต่ถ้า 2 คำสั่งขึ้นไป ต้องใส่เสมอ

```
if (เงื่อนไข)  
    คำสั่ง;  
else  
    คำสั่ง;
```

คำสั่งเดียว ไม่ต้องใส่ { } ก็ได้

```
if (เงื่อนไข) {  
    คำสั่งที่1;  
    คำสั่งที่2;  
} else {  
    คำสั่งที่3;  
    คำสั่งที่4;  
}
```

หลายคำสั่ง ให้ใส่ { } เสมอ



Style การเขียนโค้ด

```
#include <stdio.h>
void main() {
    tab int math, eng;
    printf("Enter math score :");
    scanf("%d", &math);
    printf("Enter english score :");
    scanf("%d", &eng);

    if (math>=60 && eng>=55) {
        tab printf("Pass\n");
    } else {
        tab printf("Fail\n");
    }
}
```

ปีกกาเปิดอาจเขียนไว้ในบรรทัดเดียวกัน
หรือขึ้นบรรทัดใหม่ก็ได้

เว้นบรรทัดว่างได้ เพื่อให้อ่านง่าย

หลังจากเปิดปีกกา
ควรเยื้อง 1 tab

ใส่ comment บ้างกันสิ



การจัดรูปแบบโค้ดอัตโนมัติ

คลิกขวาที่ส่วนใด ๆ ของโค้ด แล้วเลือก Format this file โปรแกรม CodeBlocks จะจัดรูปแบบให้อัตโนมัติ

The screenshot shows the CodeBlocks IDE interface. The main window displays a C program named `*main.c` with the following code:

```
1  #include <stdio.h>
2  void main()
3  {
4      int a, b, c;
5      scanf("%d %d %d", &a, &b, &c);
6      float sum = a + b + c;
7      float average = sum/3;
8
9      printf("Summation = %d", sum);
10     printf("Average = %f" average
```

A right-click context menu is open over the code, with the option "Format this file (AStyle)" highlighted. The menu items include: Switch to, Insert, Swap header/source, Edit, Bookmarks, Folding, Format this file (AStyle), Add Todo item..., Split view, Configure editor..., and Properties... The status bar at the bottom shows "Line 3, Column 2".



คำสั่งเงื่อนไข 2 ทาง แบบย่อ

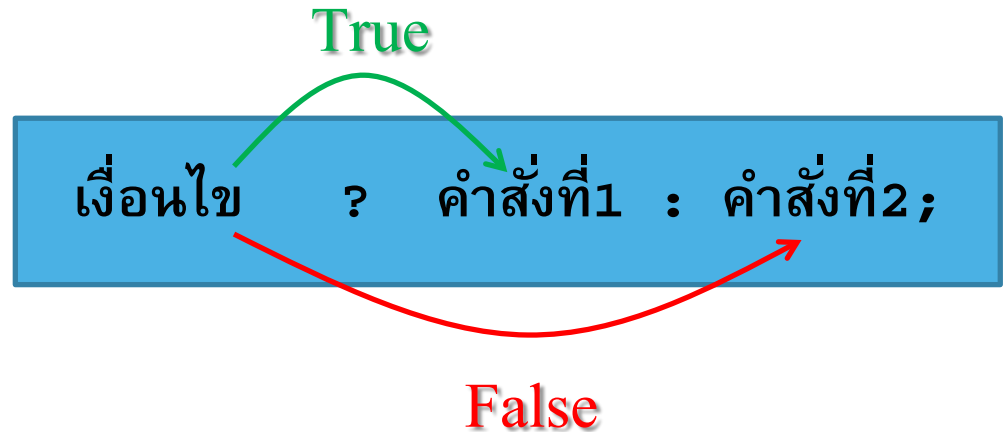
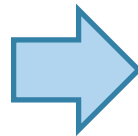
```
if (เงื่อนไข){
```

```
    คำสั่งที่1;    True
```

```
} else {
```

```
    คำสั่งที่2;    False
```

```
}
```





คำสั่งเงื่อนไข 2 ทาง แบบย่อ

```
#include <stdio.h>

void main() {
    int x1, x2;
    printf("First value> ");
    scanf("%d", &x1);
    printf("Second value> ");
    scanf("%d", &x2);

    int max;
    if (x1 > x2) {
        max = x1;
    } else {
        max = x2;
    }

    printf("Maximum is %d", max);
}
```

```
#include <stdio.h>

void main() {
    int x1, x2;
    printf("First value> ");
    scanf("%d", &x1);
    printf("Second value> ");
    scanf("%d", &x2);

    int max = (x1 > x2) ? x1 : x2;

    printf("Maximum is %d", max);
}
```



กิจกรรม

โปรแกรมคำนวณค่าโบนัส โดยพิจารณาจากประสบการณ์ในการทำงาน หากทำงานไม่ถึง 3 ปีจะได้โบนัส 1 เท่าของเงินเดือน แต่หากทำงานมาแล้วตั้งแต่ 3 ปีขึ้นไปจะได้โบนัสเป็น 2 เท่าของเงินเดือน จงเขียนโปรแกรมรับค่าเงินเดือนและประสบการณ์ทำงาน และแสดงค่าของโบนัสที่พนักงานจะได้รับ

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

```
void main() {
```

```
    int salary, experience, bonus;
```

```
    printf("Input your salary : ");
```

```
    scanf("%d", &salary);
```

```
    printf("Input experience year(s) : ");
```

```
    scanf("%d", &experience);
```

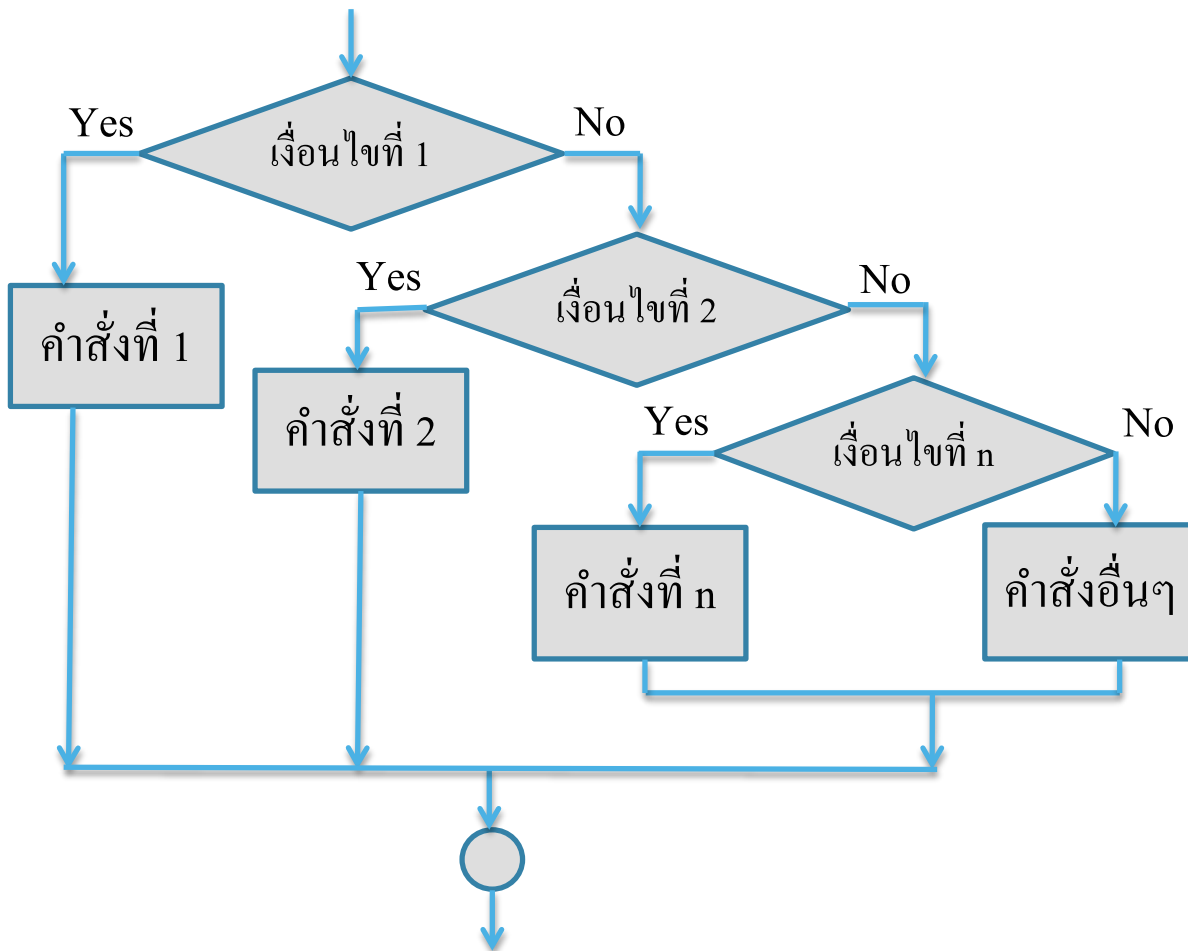
```
    bonus = (experience >= 3) ? _____ : _____;
```

```
    printf("Your bonus is %d baht", bonus);
```

```
}
```

เงื่อนไขซ้อนเงื่อนไข

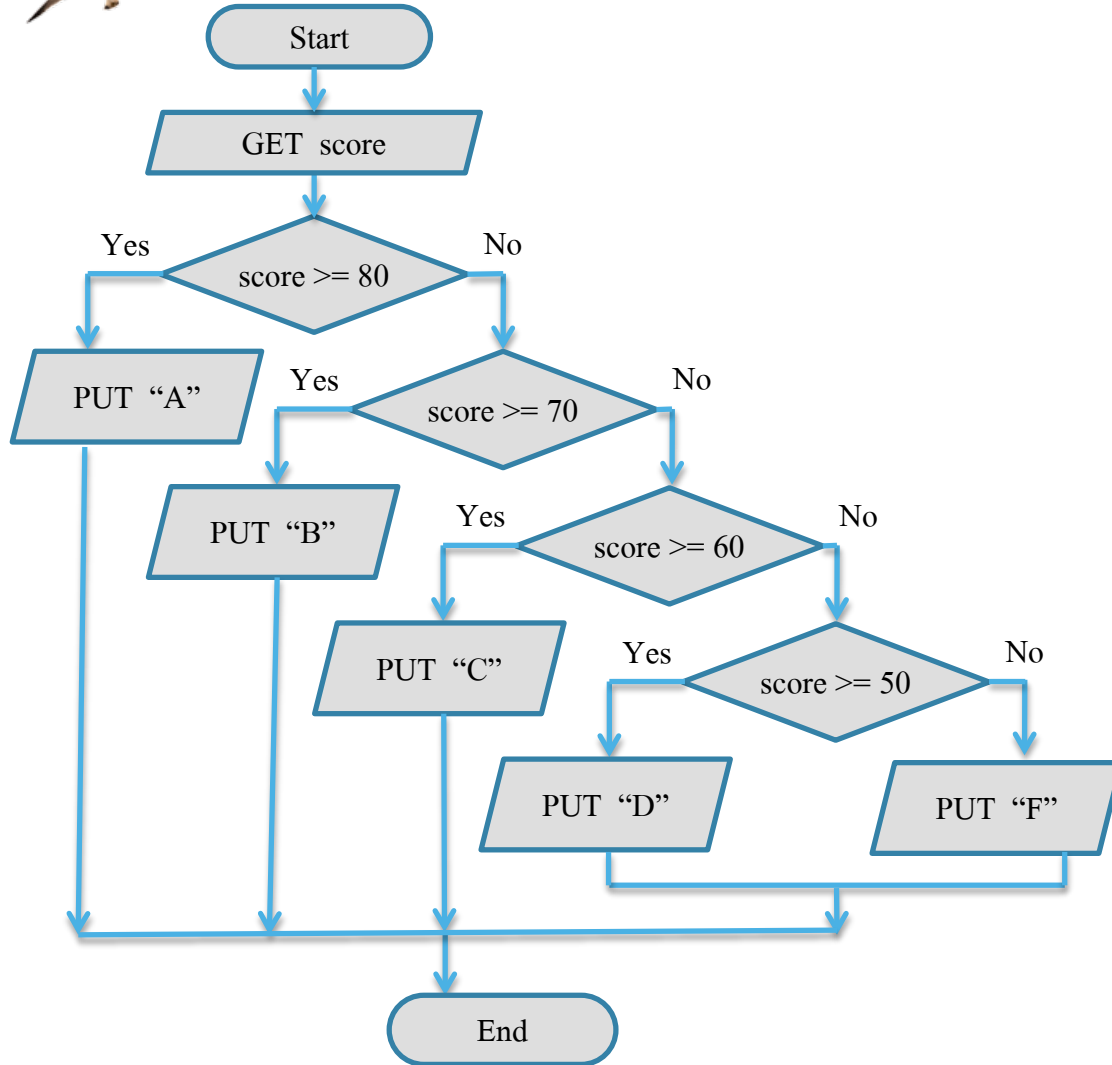
รูปแบบคำสั่งภาษาซี



```
if (เงื่อนไขที่ 1){  
    คำสั่งที่1;  
} else if (เงื่อนไขที่ 2){  
    คำสั่งที่2;  
} else if (เงื่อนไขที่ n){  
    คำสั่งที่ n;  
} else {  
    คำสั่งอื่นๆ;  
}
```



ตัวอย่าง

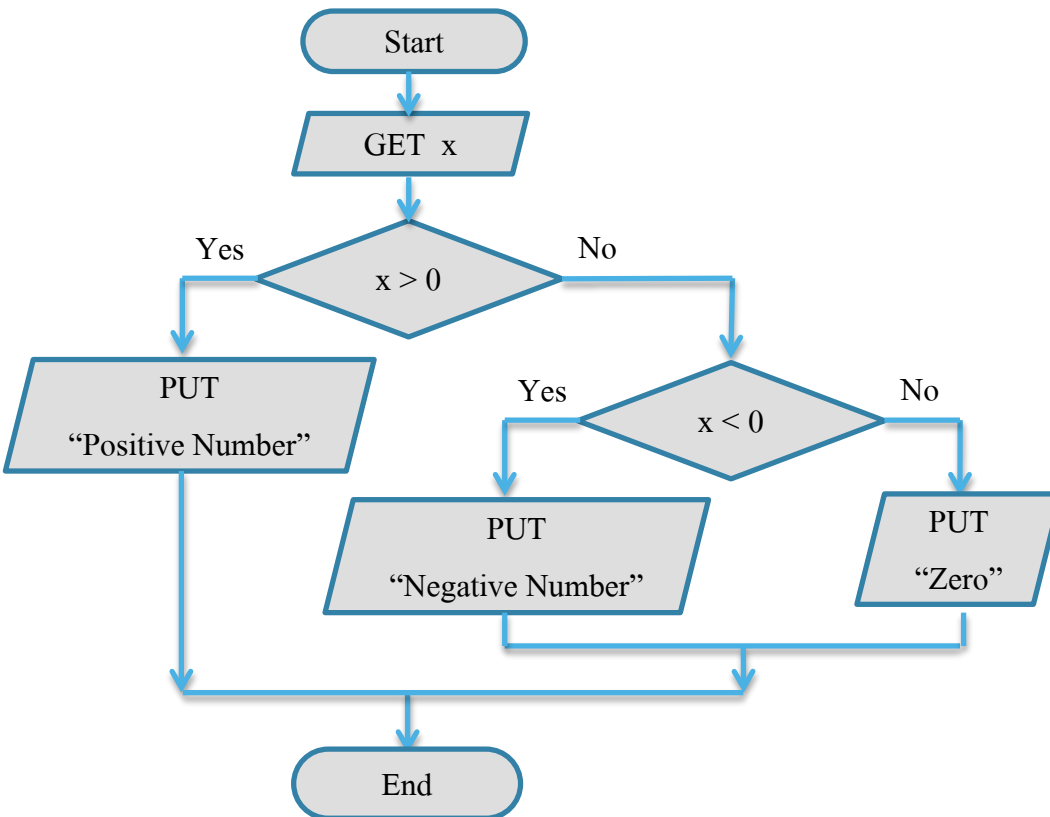


```
1 #include <stdio.h>
2 void main() {
3     int score;
4     printf("Your score:");
5     scanf("%d", &score);
6
7     if (score >= 80) {
8         printf("A\n");
9     } else if (score >= 70) {
10        printf("B\n");
11    } else if (score >= 60) {
12        printf("C\n");
13    } else if (score >= 50) {
14        printf("D\n");
15    } else {
16        printf("F\n");
17    }
18 }
```



กิจกรรม

Flowchart โปรแกรมตรวจสอบตัวเลขที่รับเข้ามาว่าเป็นบวก ลบ หรือ 0 จงเขียน
โปรแกรมภาษาซีจากขั้นตอนวิธีใน Flowchart



```
#include <stdio.h>
void main() {
    int x;
    printf("Enter a number: ");
    scanf("%d", &x);

    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
}
```



กิจกรรม

❖ ออกแบบอัลกอริทึมสำหรับ โปรแกรมร้าน U-Bar เพื่อสอบถามข้อมูลและตรวจสอบว่าลูกค้าสามารถใช้บริการร้านได้หรือไม่ โปรแกรมจะต้องรับ ชื่อ อายุ เพศ (F-ผู้หญิง, M-ผู้ชาย)

- หากอายุต่ำกว่า 18 ปี ให้บอกว่า

Baby! Go home now.

- หากอายุ 18 ปีขึ้นไป ก็ให้แสดงข้อความตามเพศ
 - ถ้าเป็นผู้ชาย บอกว่า Hi Mr.ชื่อ. Welcome to U-Bar
 - ถ้าเป็นผู้หญิง บอกว่า Hi Ms.ชื่อ. Welcome to U-Bar



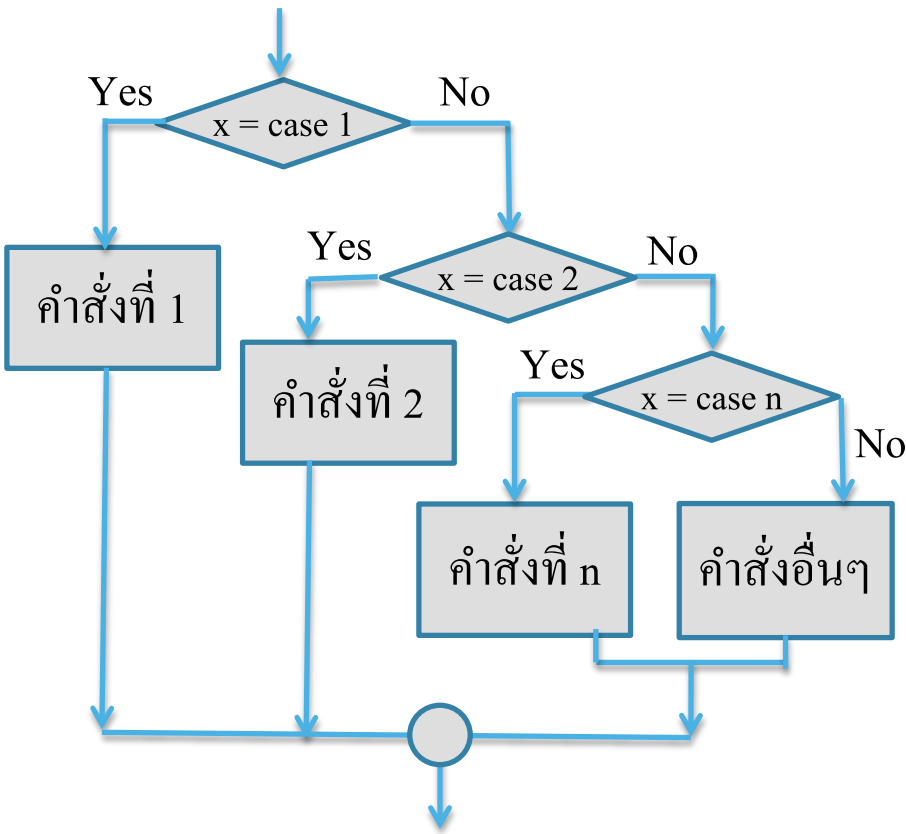
กิจกรรม

โปรแกรมคิดค่าขนส่งพัสดุ จะรับค่าน้ำหนักพัสดุนำมาเป็นกิโกรัม หลังจากนั้นคำนวณค่าส่งตามเงื่อนไขตารางด้านล่างนี้ หากค่าขนส่งที่คิดได้แล้วมีค่าตั้งแต่ 500 บาทขึ้นไป จะลดราคาค่าขนส่งลง 2% จงออกแบบอัลกอริทึมด้วย Flowchart และเขียนโปรแกรมด้วยภาษาซี

น้ำหนักพัสดุ (กก.)	ราคาต่อกิโกรัม (บาท)
< 2.5 กก.	20
2.5 – 5 กก.	15
> 5 กก.	11



คำสั่งเงื่อนไข switch-case



switch (ตัวแปรชนิด int หรือ char) {

case <ตัวเลขหรืออักขระ>: เครื่องหมาย Colon

คำสั่งที่ 1;

break;

case <ตัวเลขหรืออักขระ>: ใช้คำสั่ง break เพื่อ

คำสั่งที่ 2;

break;

ป้องกันการทำงานใน case ถัดไป

case <ตัวเลขหรืออักขระ>:

คำสั่งที่ n;

break;

default:
คำสั่งอื่นๆ;

ถ้าค่าของตัวแปรไม่ตรงกับค่าใน case ใดๆ จะทำคำสั่งภายใต้ default

}



ตัวอย่าง

โปรแกรมตรวจสอบตัวเลขที่รับเข้ามาว่าเป็นเลขคู่ (even) หรือเลขคี่ (odd) โดยใช้ switch-case

```
1 #include <stdio.h>
2
3 void main() {
4     int x;
5     printf("Enter a number: ");
6     scanf("%d", &x);
7
8     switch (x%2) {
9         case 0:
10            printf("Even Number");
11            break;
12        case 1:
13            printf("Odd Number");
14            break;
15    }
16 }
```



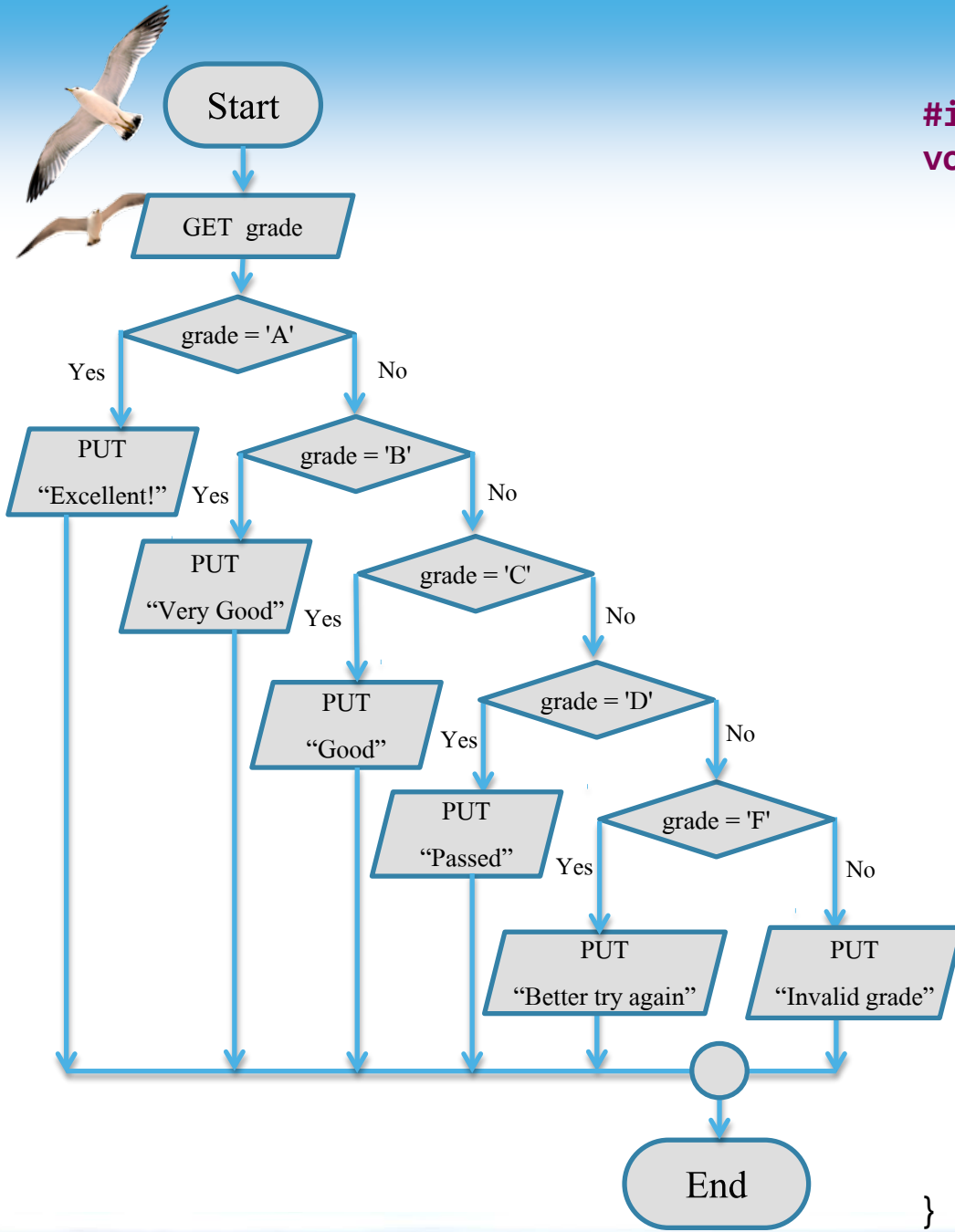
switch-case กับ เงื่อนไขซ้อนเงื่อนไข

```
#include <stdio.h>
void main () {
    char grade;
    printf("What is your grade ?\n");
    scanf("%c", &grade);

    switch(grade) {
        case 'A' : -----
            printf("Excellent!\n" );
            break;
        case 'B' : -----
            printf("Very Good\n" );
            break;
        case 'C' : -----
            printf("Good\n" );
            break;
        case 'D' : -----
            printf("Passed\n" );
            break;
        case 'F' : -----
            printf("Better try again\n" );
            break;
        default : -----
            printf("Invalid grade\n" );
    }
}
```

```
#include <stdio.h>
void main () {
    char grade;
    printf("What is your grade ?\n");
    scanf("%c", &grade);

    •if (grade == 'A') {
        printf("Excellent!\n" );
    }
    •} else if (grade == 'B') {
        printf("Very Good\n" );
    }
    •} else if (grade == 'C') {
        printf("Good\n" );
    }
    •} else if (grade == 'D') {
        printf("Passed\n" );
    }
    •} else if (grade == 'F') {
        printf("Better try again\n" );
    }
    •} else {
        printf("Invalid grade\n" );
    }
}
```



```

#include <stdio.h>
void main () {
    char grade;
    printf("What is your grade ?\n");
    scanf("%c", &grade);

    switch(grade) {
        case 'A' :
            printf("Excellent!\n" );
            break;
        case 'B' :
            printf("Very Good\n" );
            break;
        case 'C' :
            printf("Good\n" );
            break;
        case 'D' :
            printf("Passed\n" );
            break;
        case 'F' :
            printf("Better try again\n" );
            break;
        default :
            printf("Invalid grade\n" );
    }
}
  
```



กิจกรรม

❖ เมื่อลบคำสั่ง `break` ในทุกๆ case ผลลัพธ์เมื่อ input ค่าเป็น B จะมี output คืออะไร



กิจกรรม

จงหาผลลัพธ์ของโปรแกรมต่อไปนี้

```
#include <stdio.h>
void main() {
    int k = 2;
    int p = 1;
    switch(p) {
        case 1: k++;
        case 2: k++;
        case 3: k++;
        default: k++;
    }
    printf("%d\n", k);
}
```



switch-case แบบรวมหลาย case

โปรแกรมแสดงชื่อฤดู โดยรับค่าเดือนที่เป็นตัวเลข และแสดงชื่อฤดู โดยกำหนดให้มีหลาย case ในแต่ละฤดู และเปรียบเทียบการเขียนโปรแกรมที่ทำงานเหมือนกันด้วยการใช้เงื่อนไขซ้อนเงื่อนไข

```
1 #include <stdio.h>
2 void main() {
3     int month;
4     scanf("%d", &month);
5
6     switch (month) {
7         case 11: case 12: case 1: case 2:
8             printf("Winter\n");
9             break;
10        case 3: case 4: case 5:
11            printf("Summer\n");
12            break;
13        case 6: case 7: case 8: case 9: case 10:
14            printf("Rainy\n");
15            break;
16
17        default :
18            printf("No season\n");
19        }
20 }
```

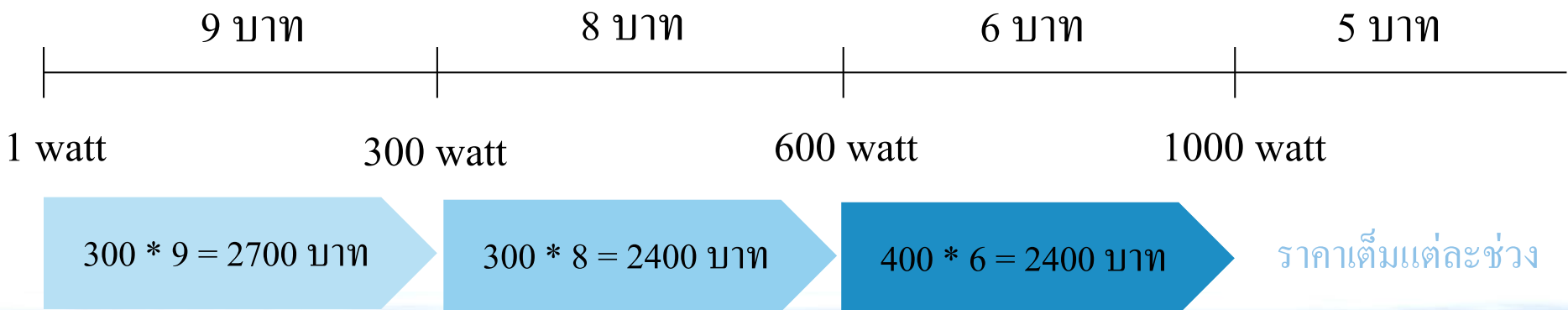
```
1 #include <stdio.h>
2 void main() {
3     int month;
4     scanf("%d", &month);
5
6
7     if (month==11 || month==12 || month==1 || month==2)
8         printf("Winter\n");
9
10    else if (month==3 || month==4 || month==5)
11        printf("Summer\n");
12
13    else if (month==6 || month==7 || month==8 || month==9
14            || month==10)
15        printf("Rainy\n");
16
17    else
18        printf("No season\n");
19
20 }
```




โปรแกรมคิดค่าไฟฟ้า

เป็นโปรแกรมสำหรับคำนวณค่าไฟฟ้าโดยรับจำนวนวัตต์ที่ถูกค่าใช้ และคำนวณค่าไฟฟ้าตามจำนวนที่ใช้ หากใช้มากเกินไปที่ระบุจะคิดในอัตราต่อวัตต์ที่แตกต่างกัน ดังนี้

- 9 บาท/วัตต์ สำหรับ 300 วัตต์แรก
- 8 บาท/วัตต์ สำหรับ 301 วัตต์ถัดไปจนถึง 600 วัตต์
- 6 บาท/วัตต์ สำหรับ 601 วัตต์ถัดไปจนถึง 1000 วัตต์
- 5 บาท/วัตต์ สำหรับค่าไฟฟ้าที่เกิน 1000 วัตต์





ตัวอย่างการคำนวณค่าไฟในช่วงต่างๆ

9 บาท

8 บาท

6 บาท

5 บาท

1 watt

300 watt

600 watt

1000 watt

$$300 * 9 = 2700 \text{ บาท}$$

$$300 * 8 = 2400 \text{ บาท}$$

$$400 * 6 = 2400 \text{ บาท}$$

ราคาเต็มแต่ละช่วง

ใช้ไฟ 50 watt คิดเป็น $50 * 9 = 450$ บาท

ใช้ไฟ 301 watt คิดเป็น $2700 + ((301-300) * 8) = 2708$ บาท

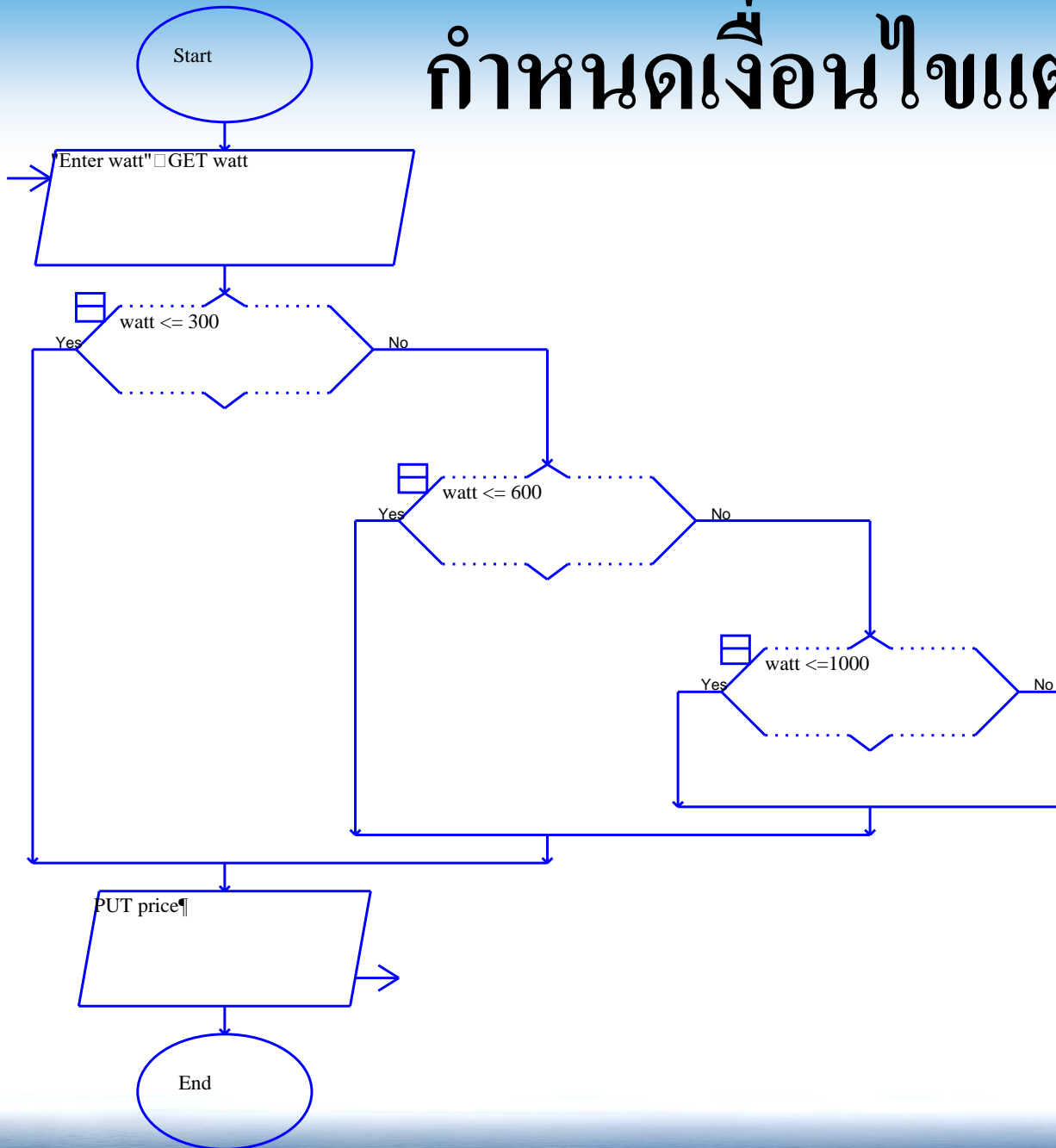
ใช้ไฟ 650 watt คิดเป็น $2700 + 2400 + ((650-600) * 6) = 5400$ บาท

ใช้ไฟ 1010 watt คิดเป็น $2700 + 2400 + 2400 + ((1010-1000) * 5) = 7750$ บาท

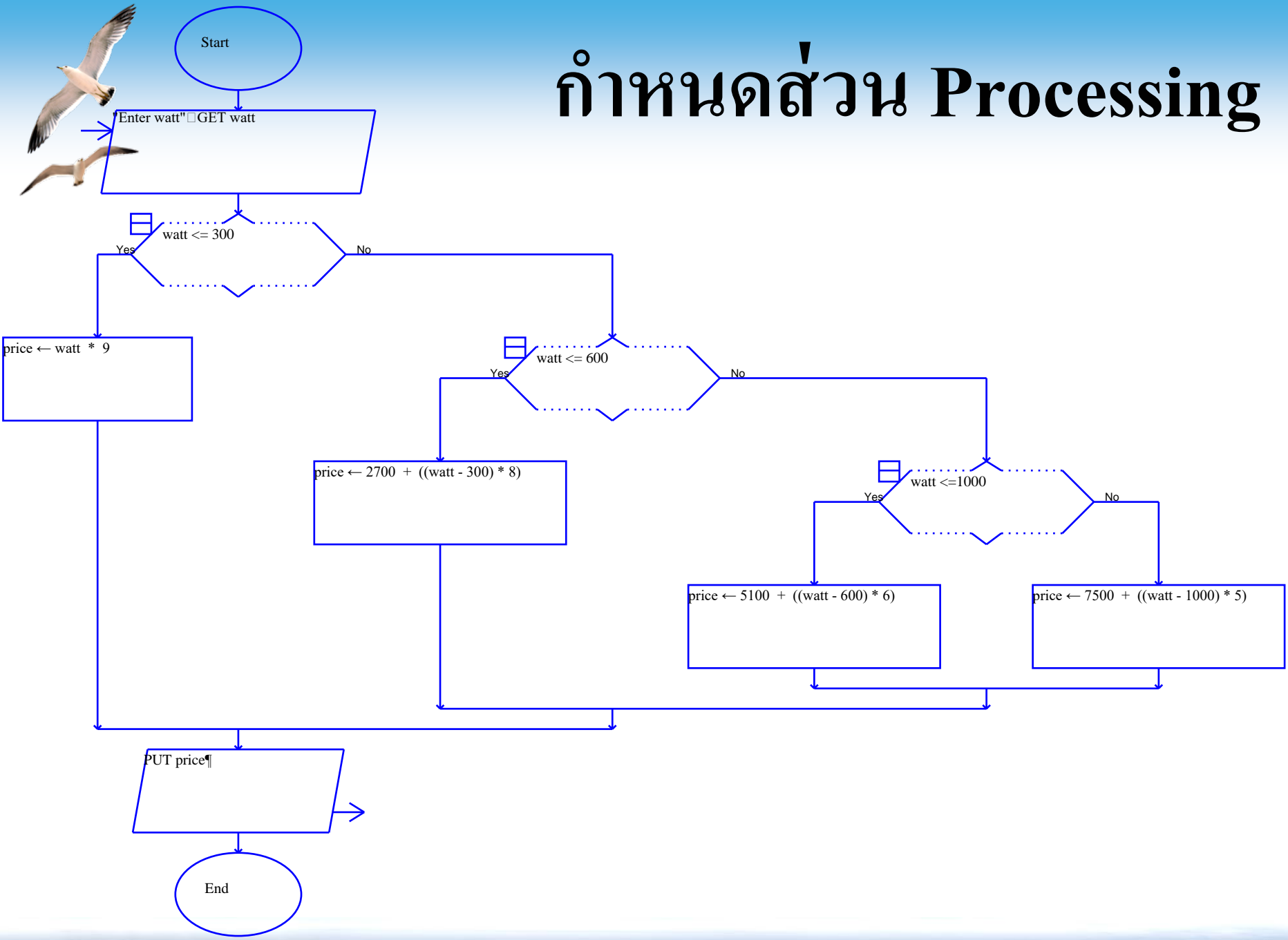
input



กำหนดเงื่อนไขในแต่ละช่วง



กำหนดส่วน Processing





สร้างเป็นโปรแกรมจริง

```
#include<stdio.h>
void main(){
    int watt, price;
    printf("Watt Used : ");
    scanf("%d",&watt);

    if (watt<=300) {
        price = watt * 9;
    } else if (watt<=600) {
        price = 2700 + ((watt-300)*8);
    } else if (watt<=1000) {
        price = 5100 + ((watt-600)*6);
    } else {
        price = 7500 + ((watt-1000)*5);
    }
    printf("Electric value : %d", price);
}
```



5 วิธีในการเรียนเขียนโปรแกรมอย่างรวดเร็ว

1. ออกแบบอัลกอริทึมก่อนเริ่มเขียนโปรแกรม โดยหาโจทย์ปัญหามาแตกเป็นขั้นตอน ก่อนจะเริ่มเขียนโค้ด
2. ศึกษาตัวอย่างโค้ด ทำความเข้าใจ และลองคิดตามว่าแต่ละบรรทัดมีการทำงานอย่างไร แล้วลองเขียนซ้ำโดยไม่ต้องดูโค้ดตัวอย่าง
3. ทดสอบพิมพ์โค้ดเอง ไม่ copy/paste เพื่อให้ตนเองเรียนรู้ไวยากรณ์ของภาษาโปรแกรมแบบอัตโนมัติ
4. เรียนรู้การใช้เครื่องมือ Debug – เพื่อดูผลที่เกิดจากการทำงานแต่ละบรรทัด
5. หาแหล่งความรู้เพิ่มเติม – ค้นใน google, หาคลิปสอนใน YouTube