

การเขียนโปรแกรมโดยใช้ภาษาสัญลักษณ์

Basic Decision flowchart

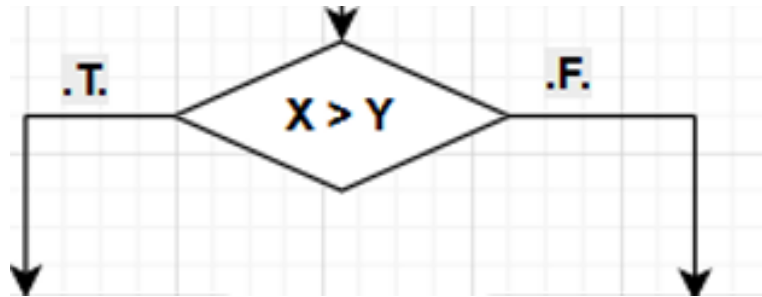
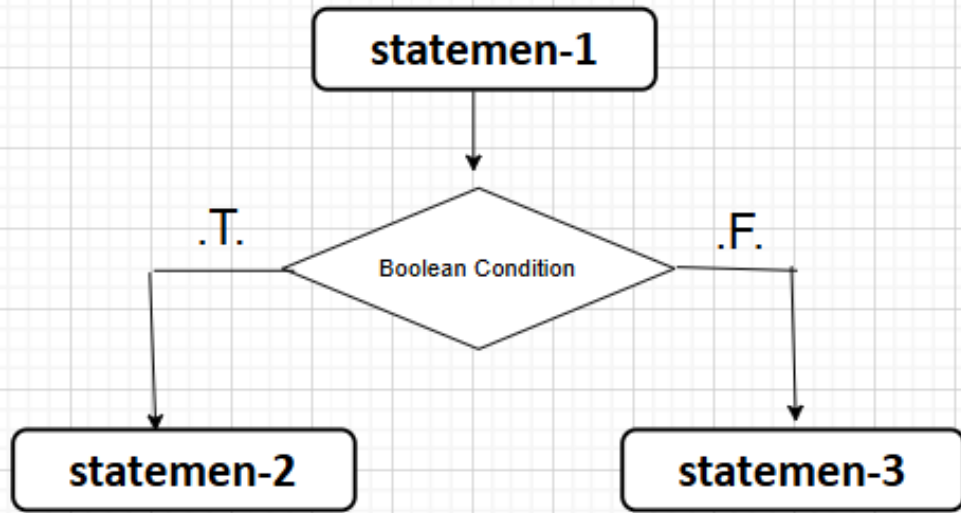
Flowchart

- เครื่องมือในการบรรยายการทำงานของ **statement** ของโปรแกรม

- Symbol

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Boolean Datatype: TRUE FALSE



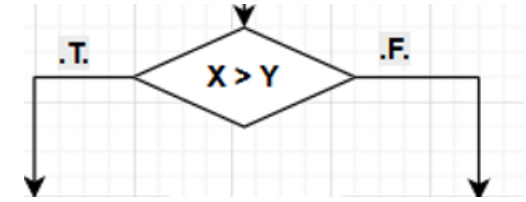
main.pas

```
1 program CalculateNet;  
2 var  
3   flag: boolean;  
4 begin  
5   flag := true;  
6   writeln( 'flag = ', flag);  
7   flag := false;  
8   writeln( 'flag = ', flag);  
9  
10  // readln( flag ); { compile error }  
11  
12 end.  
13
```

boolType.pas

Pascal ไม่สามารถใช้คำสั่ง read() / readln() กับ data boolean ได้

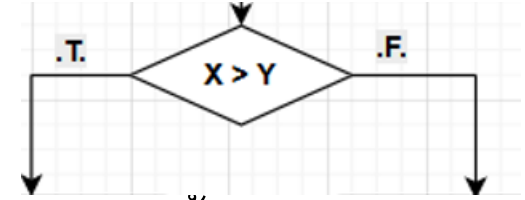
Relation Expression



- **Relation expression**) ใช้เปรียบเทียบค่าระหว่างตัวแปรหรือตัวดำเนินการ 2 ตัวขึ้นไป
- ผลลัพธ์เป็น ค่าความจริง (Boolean) คือ **true** หรือ **false**

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
=	เท่ากันหรือไม่	5 = 5	true
<>	ไม่เท่ากันหรือไม่	5 <> 3	true
>	มากกว่า	7 > 2	True
<	น้อยกว่า	2 < 4	True
>=	มากกว่าหรือเท่ากับ	5 >= 5	True
<=	น้อยกว่าหรือเท่ากับ	3 <= 2	false

Relation Expression



- **Relation expression**) ใช้เปรียบเทียบค่าระหว่างตัวแปรหรือตัวดำเนินการ 2 ตัวขึ้นไป
- ผลลัพธ์เป็น ค่าความจริง (Boolean) คือ **true** หรือ **false**
- การเขียน **relation expression** ในภาษาโปรแกรมจะสามารถเขียนในลักษณะนี้ได้:

$X > Y > Z$

ต้องเขียนเป็น

$X > Y \text{ and } Y > Z$

โดย **and** ทำหน้าที่เชื่อม relational expression

$X > Y$ กับ $Y > Z$

โอเปอเรเตอร์ **AND** เรียกว่า **Boolean expression**

Boolean Expression

- ทำหน้าที่เชื่อม relation expression
- ผลลัพธ์เป็น ค่าความจริง (Boolean) คือ true หรือ false

$x = 6$

$y = 10$

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
and	และ (and, &&)	$x > 5$ and $y < 10$	false
or	หรือ (or,)	$x = 0$ or $y = 0$	false
not	ไม่ (not, !)	not($x = 5$)	True

x	y	$x \&\& y$
T (1)	T (1)	T (1)
T (1)	F (0)	F (0)
F (0)	T (1)	F (0)
F (0)	F (0)	F (0)

x	y	$x \parallel y$
T (1)	T (1)	T (1)
T (1)	F (0)	T (1)
F (0)	T (1)	T (1)
F (0)	F (0)	F (0)

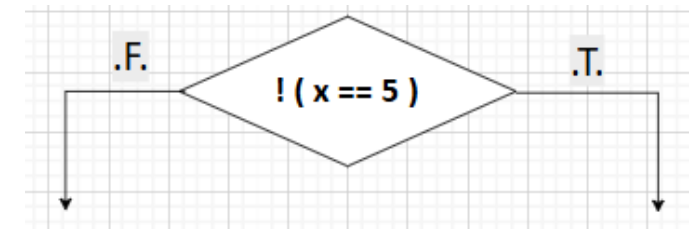
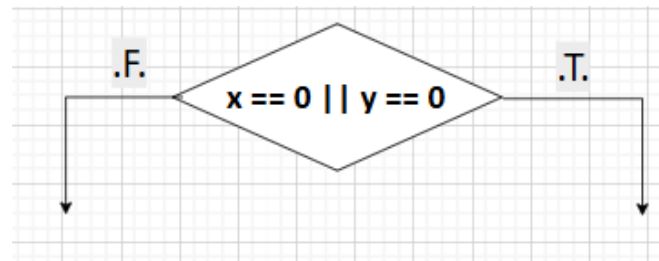
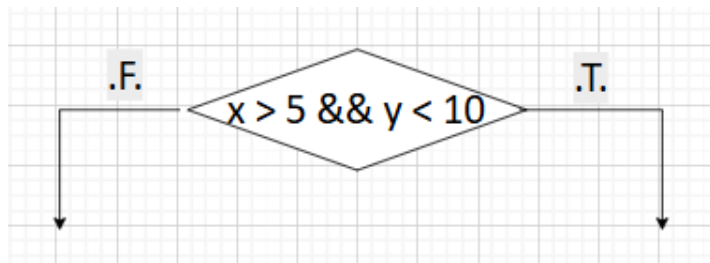
x	! x
T (1)	F (0)
F (0)	T (1)

Boolean Expression

x = 6
y = 10

- ทำหน้าที่เชื่อม relation expression
- ผลลัพธ์เป็น ค่าความจริง (Boolean) คือ true หรือ false

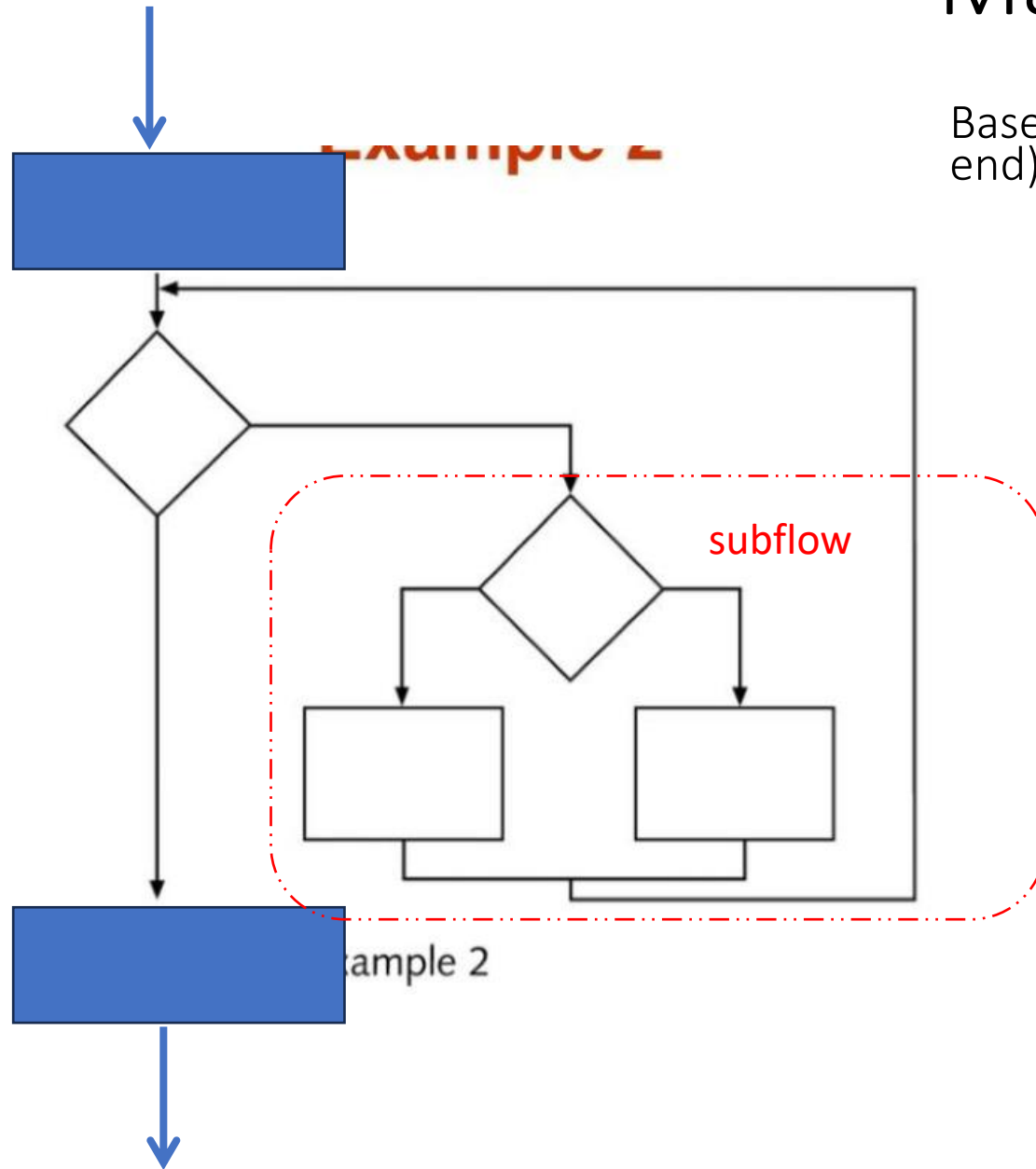
สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
and	และ (&&)	x > 5 and y < 10	false
or	หรือ ()	x = 0 or y = 0	false
not	ไม่ (!)	not(x == 5)	True



หลักการเขียน **decision flow**

Main flow and sub flow

Based on sequence execution flow (from start to end) ,the Decision symbol makes sub flows



NONSTRUCTURE FLOW

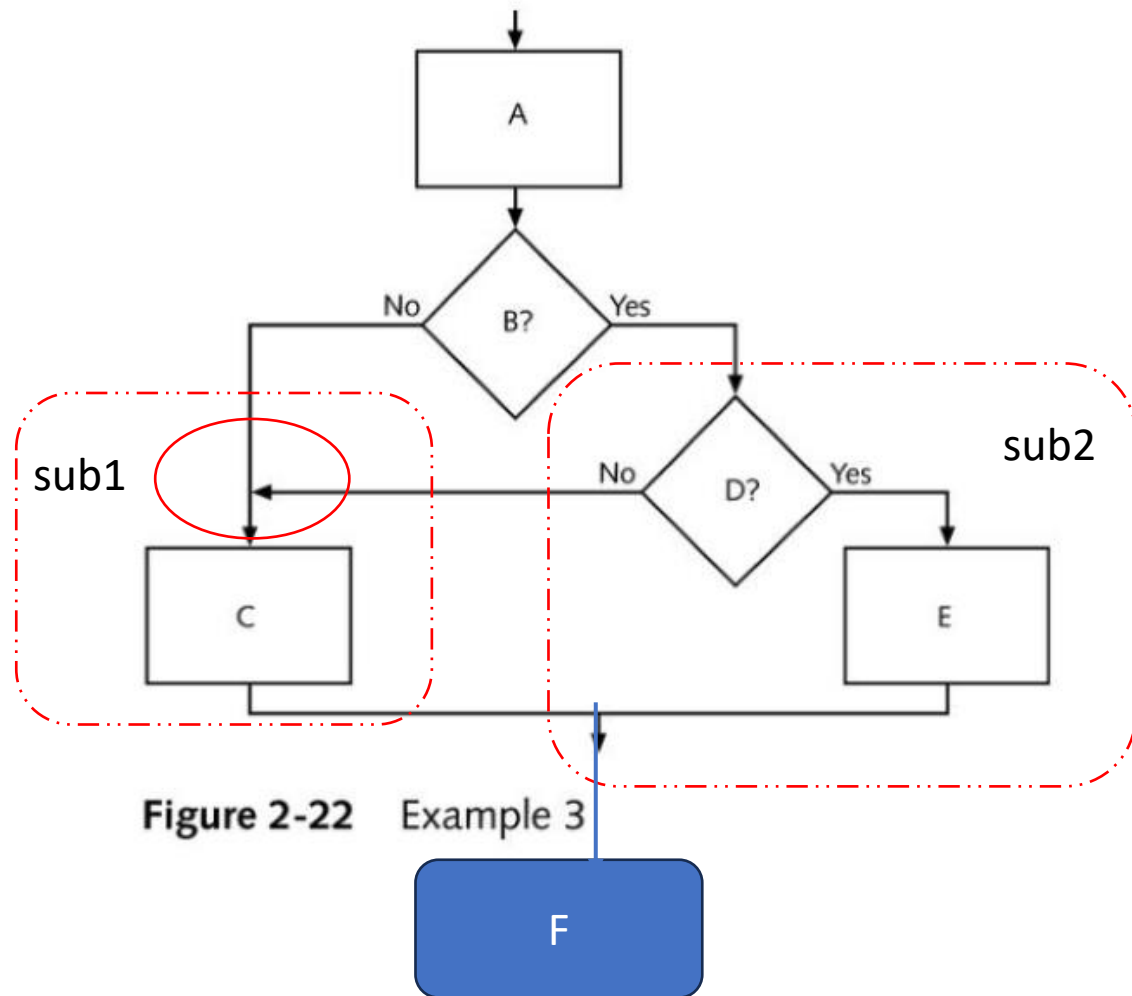


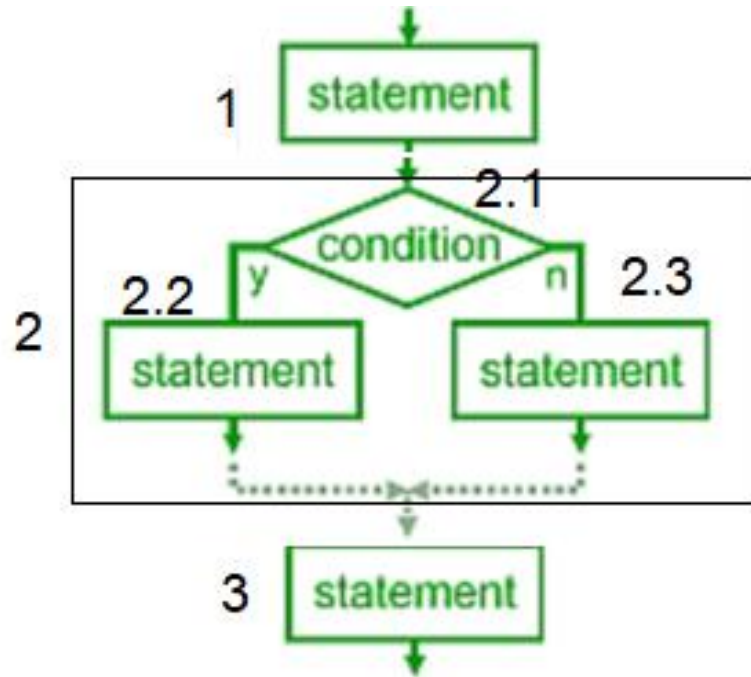
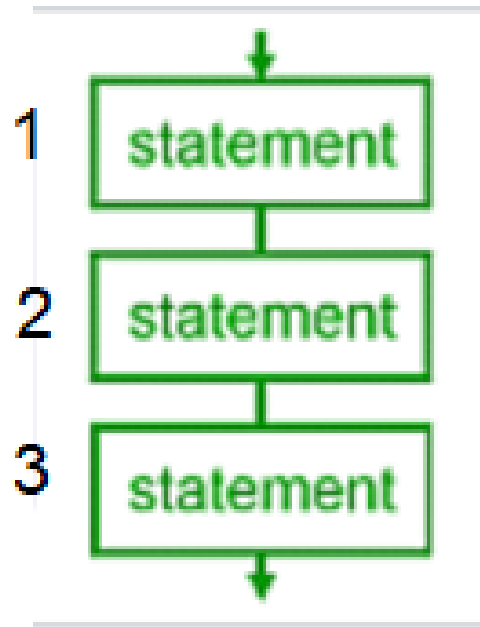
Figure 2-22 Example 3

เขียนแบบนี้ เพราะไม่สามารถแทนได้ด้วย คำสั่ง
ใน High level language ได้
เนื่องจาก sub1 มี flow ที่มาจาก sub2 ที่อยู่ที่ level
เดียวกัน

Block c mainly flows from Dimond B?

Structure Programming Concepts : Decision flow

- Based on CPU execution flow ; each of logical statement has a single entry and single exist point (หมายถึง decision flow คือ 1 logical statement)



หลักการเขียน **decision flow**

NONSTRUCTURE FLOW

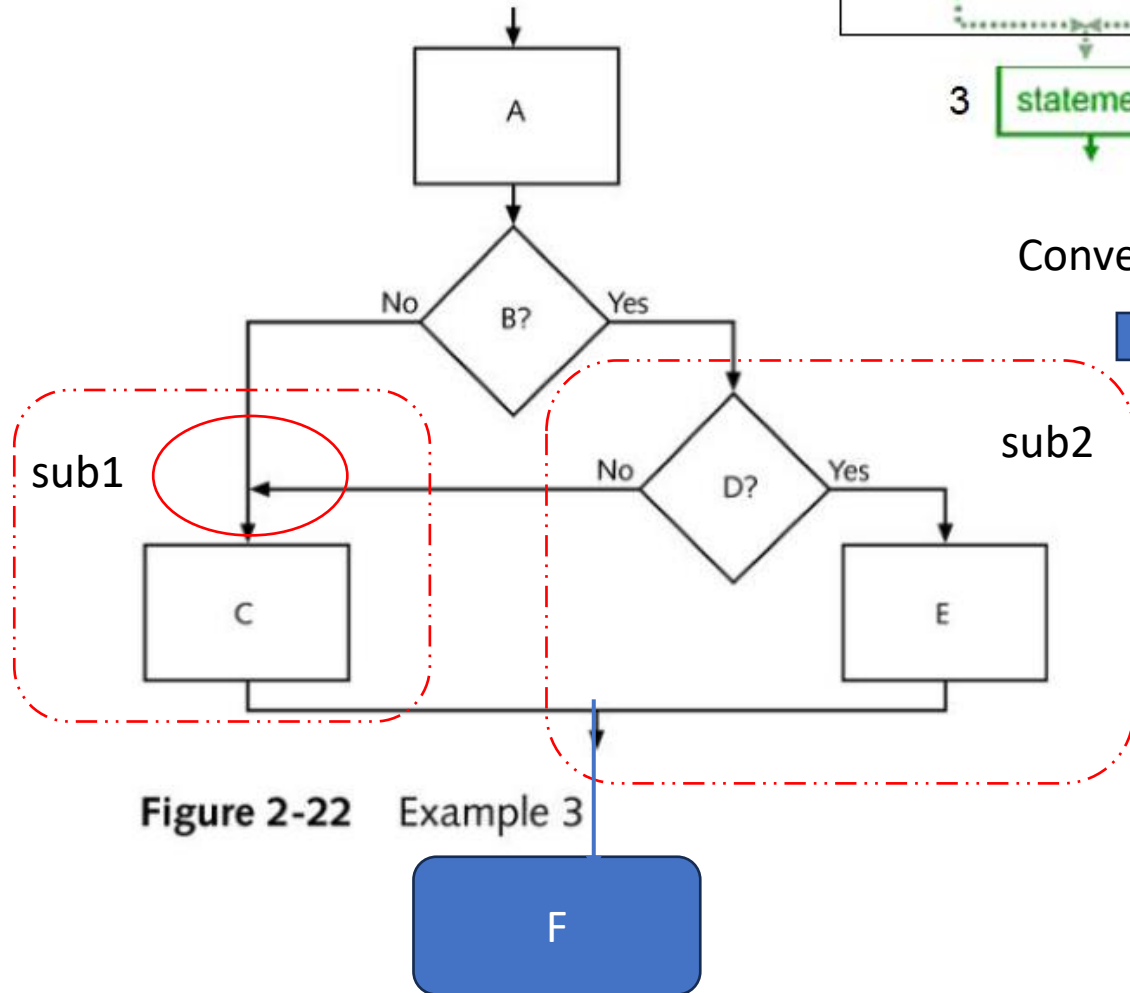
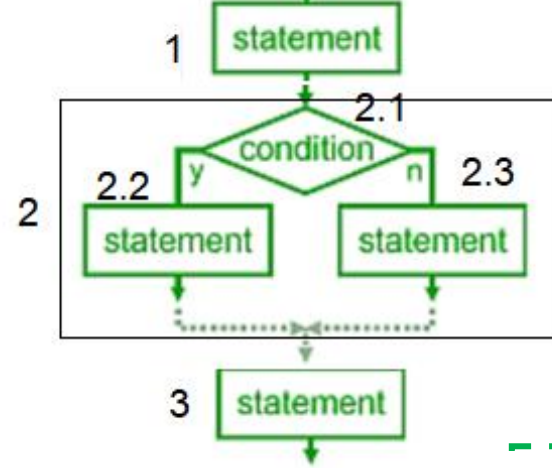


Figure 2-22 Example 3

Convert to

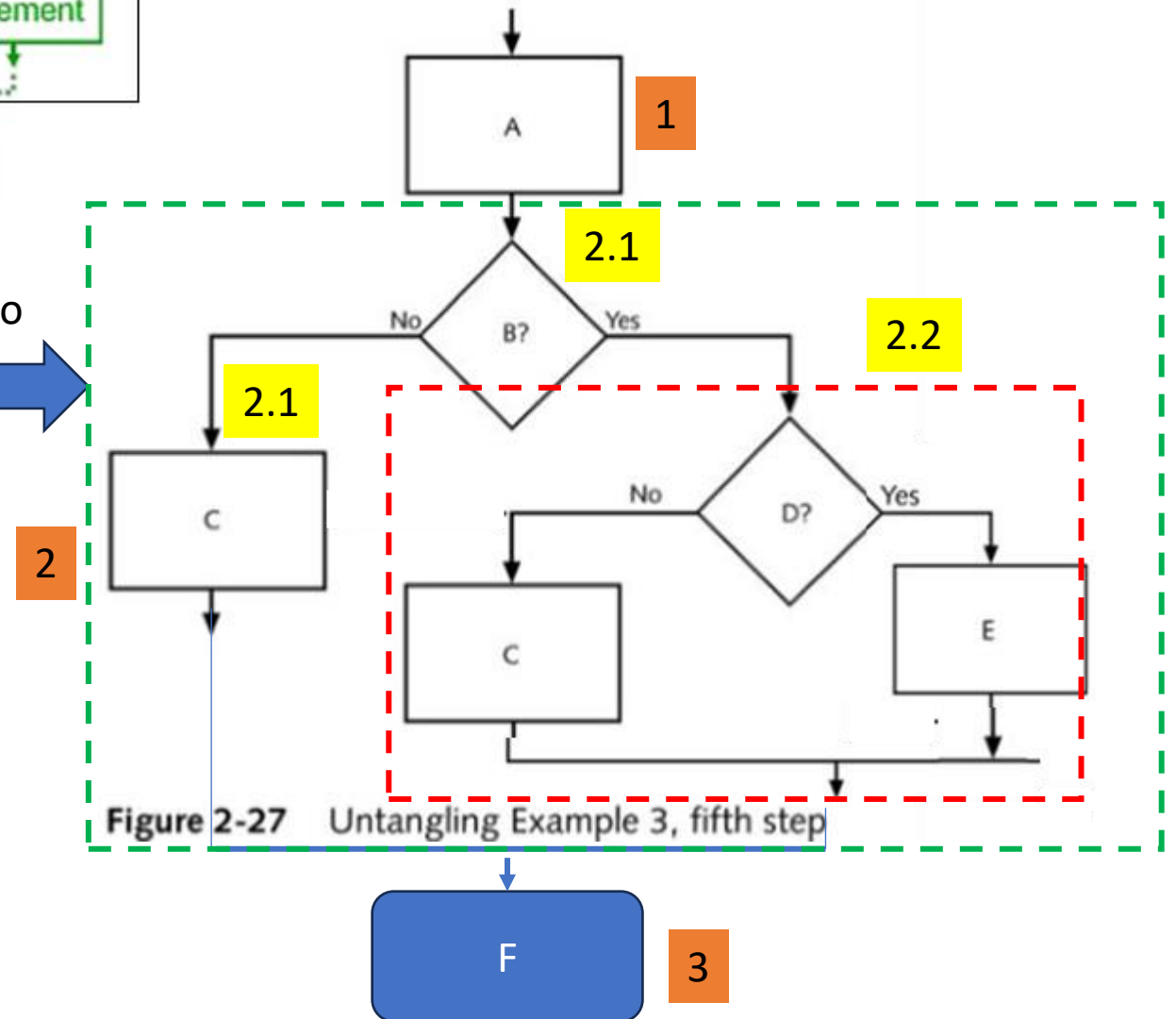


Figure 2-27 Untangling Example 3, fifth step

Block c mainly flows from Dimond B?

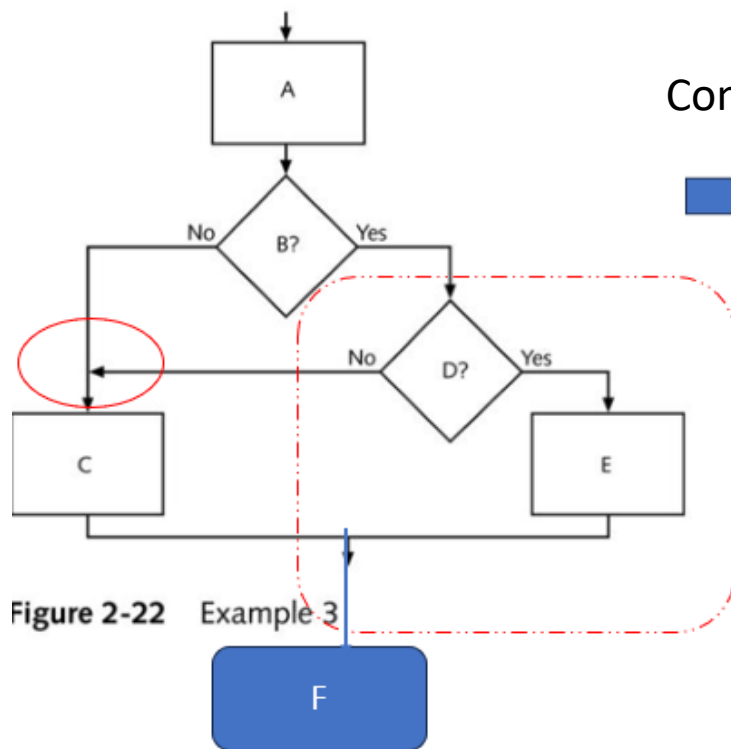
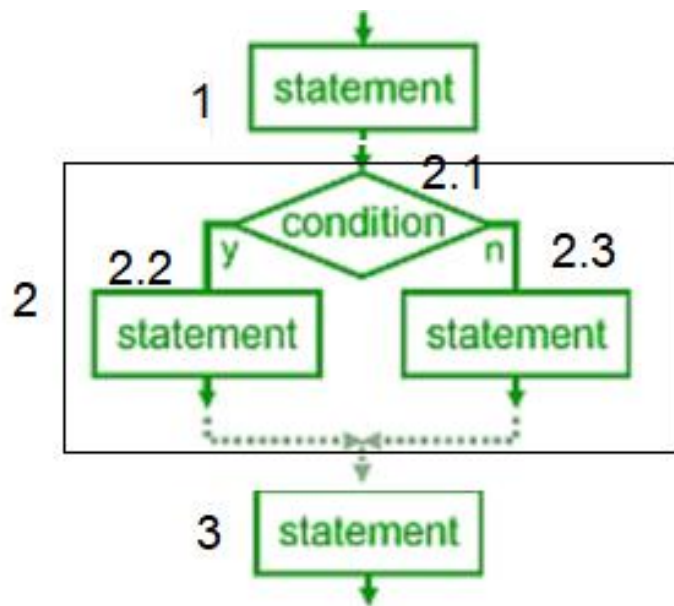


Figure 2-22 Example 3

Convert to

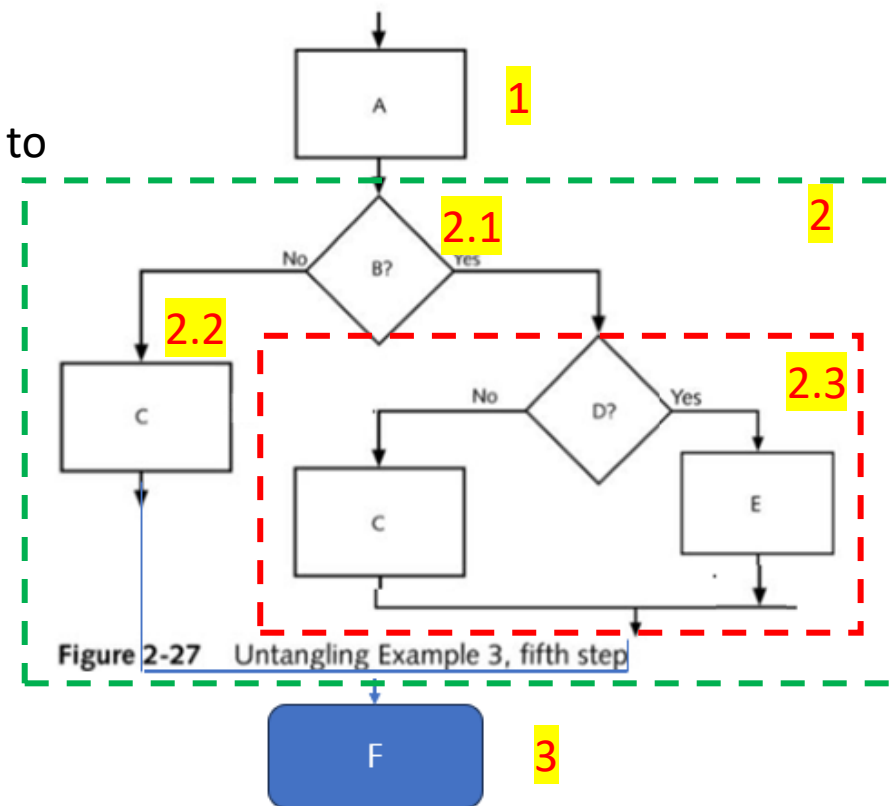
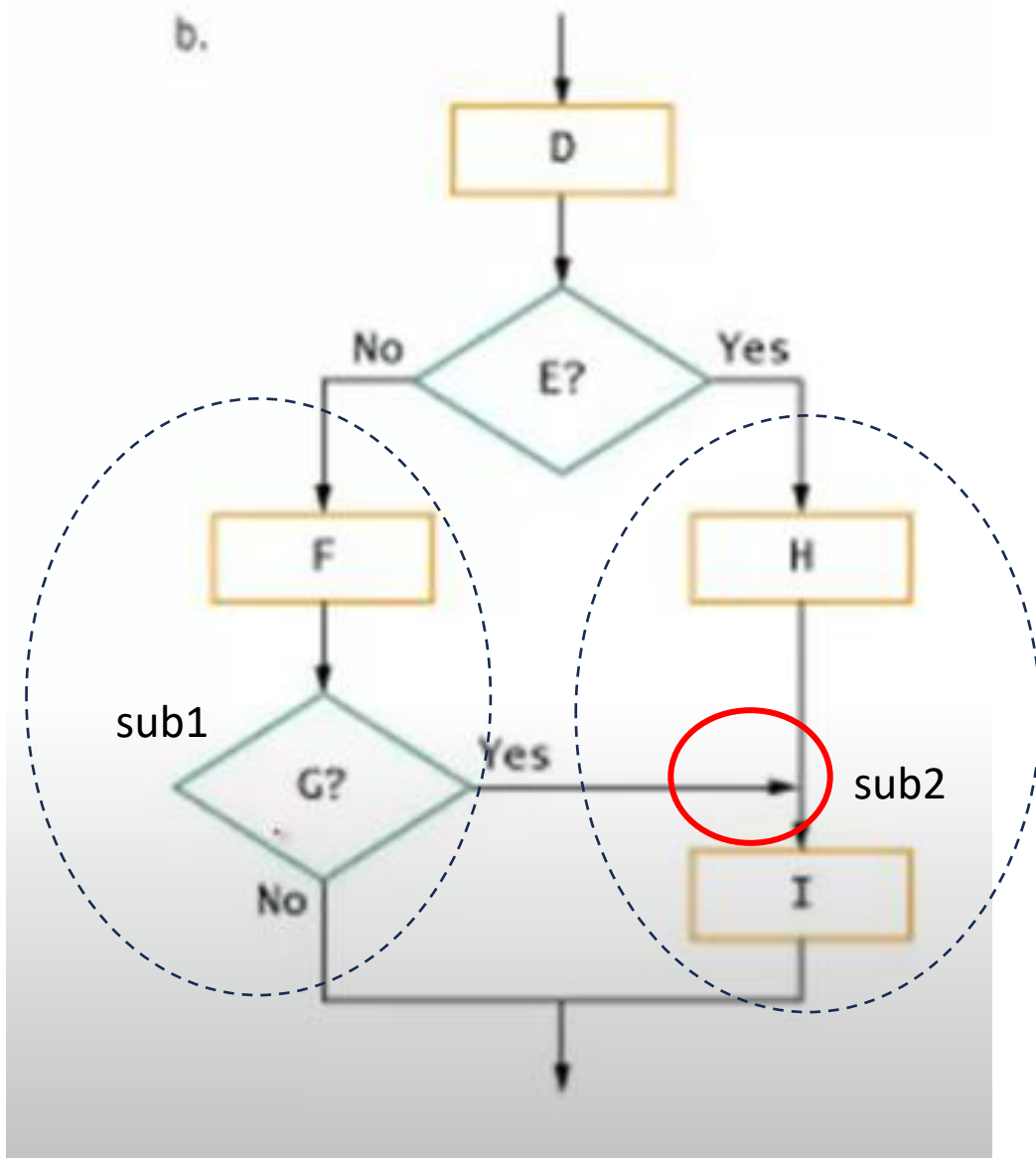


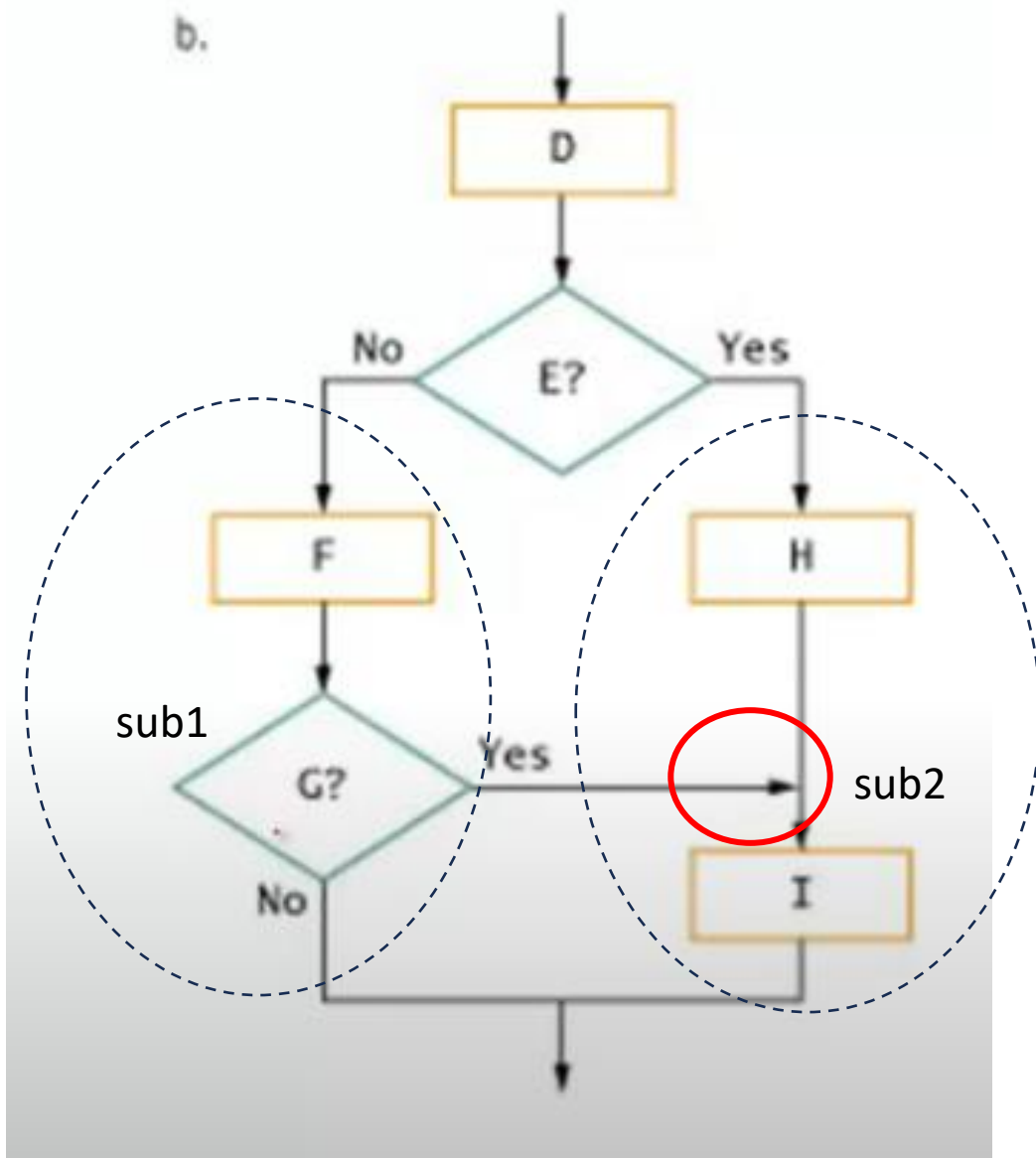
Figure 2-27 Untangling Example 3, fifth step

หลักการเขียน decision flow

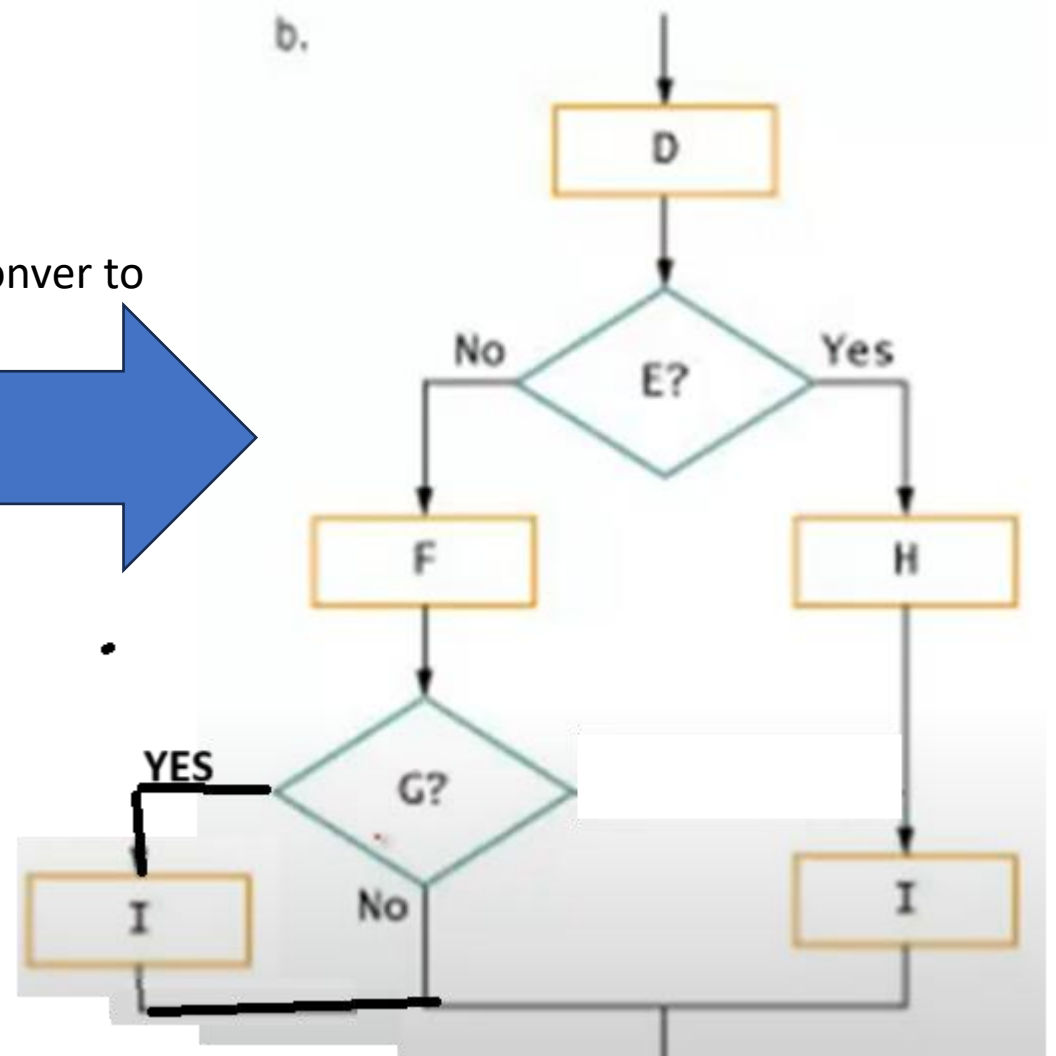
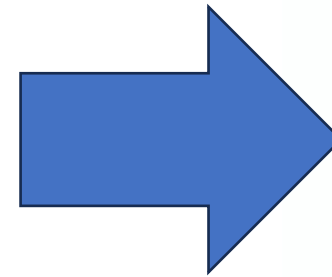


เขียนแบบนี้ เพราะไม่สามารถแทนได้ด้วย คำสั่ง
ใน High level language

หลักการเขียน decision flow

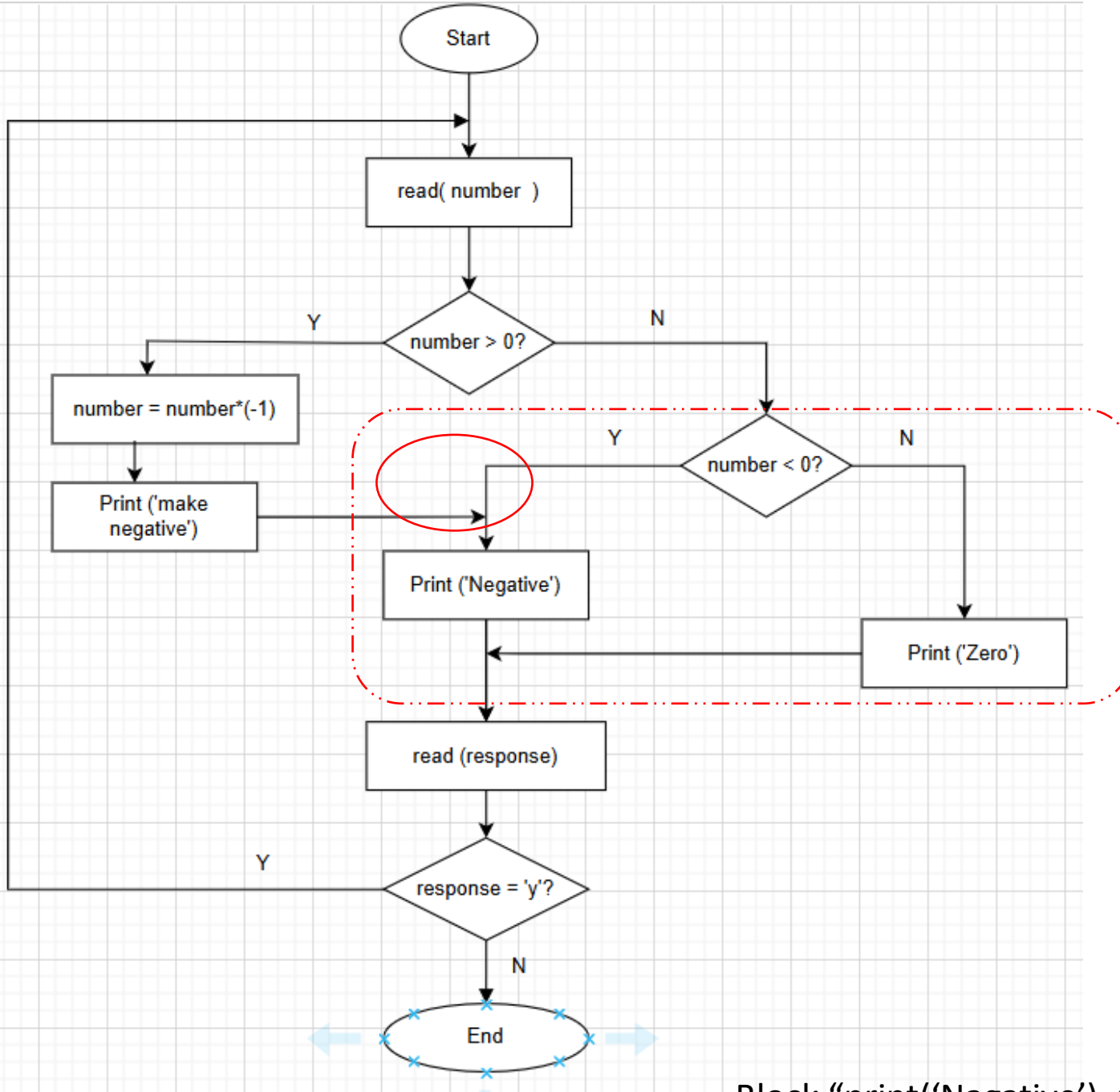


Conver to



NONSTRUCTURE FLOW

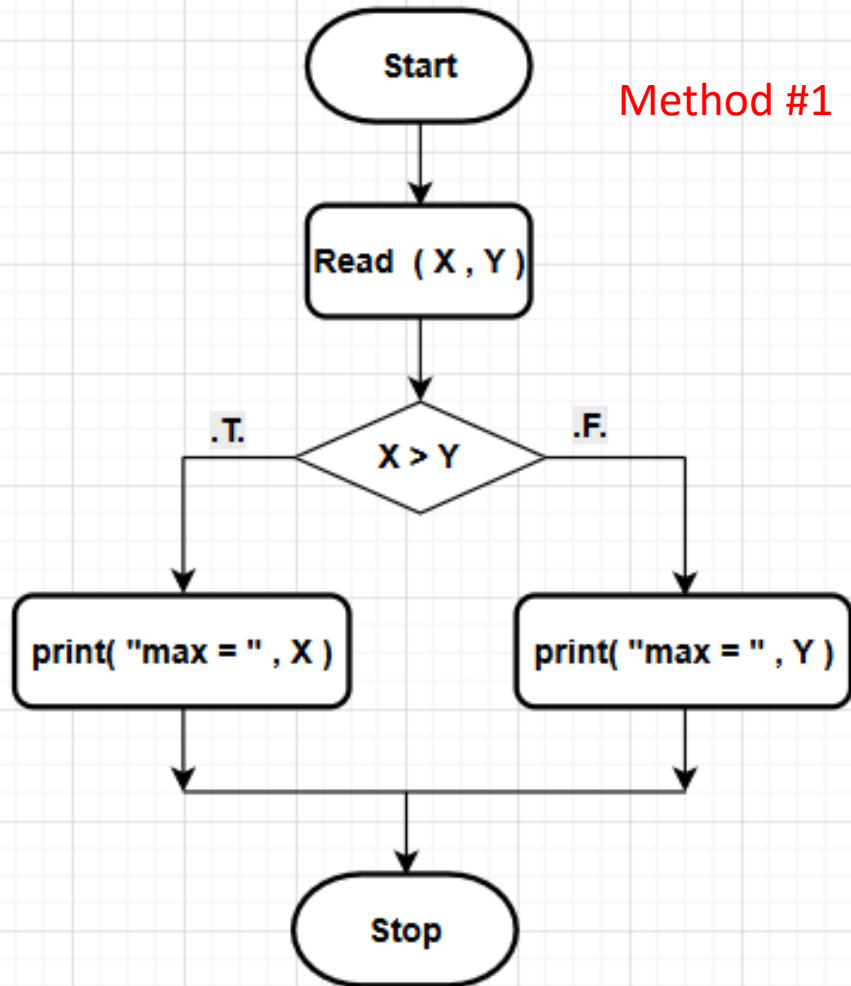
ให้นศ. Rewrite flow



Block “print('Negative')” mainly flow Dimond number <? flow

ฝึกอ่าน Flow

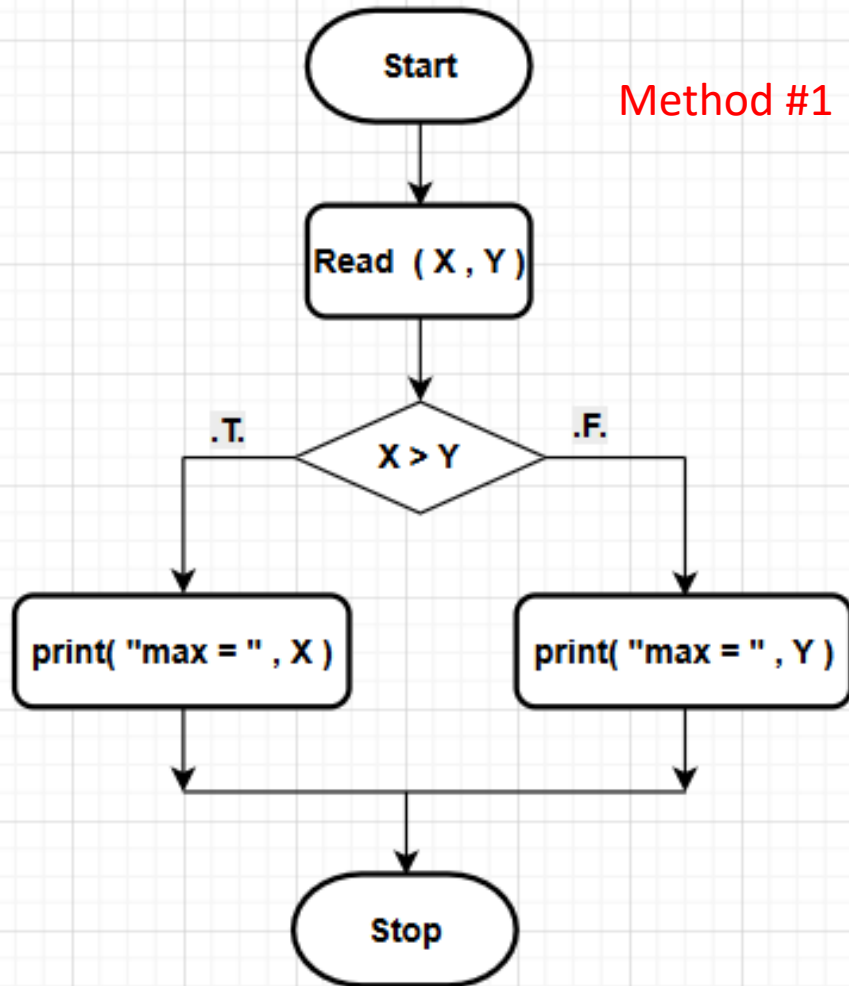
Method #1



โปรแกรมทำอะไร ?

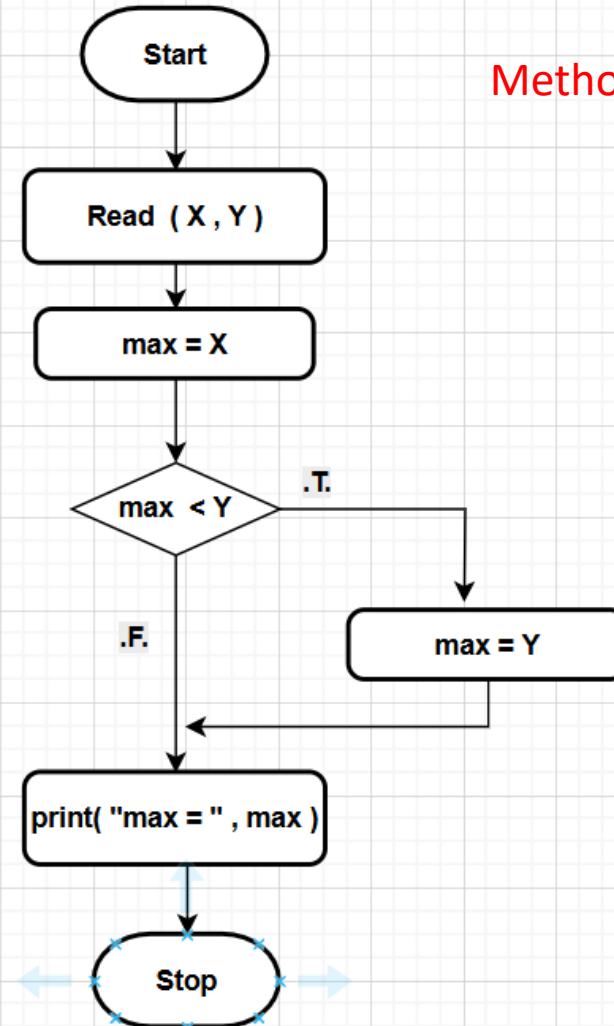
ฝึกอ่าน Flow : หาค่า max 2 ค่า

Method #1

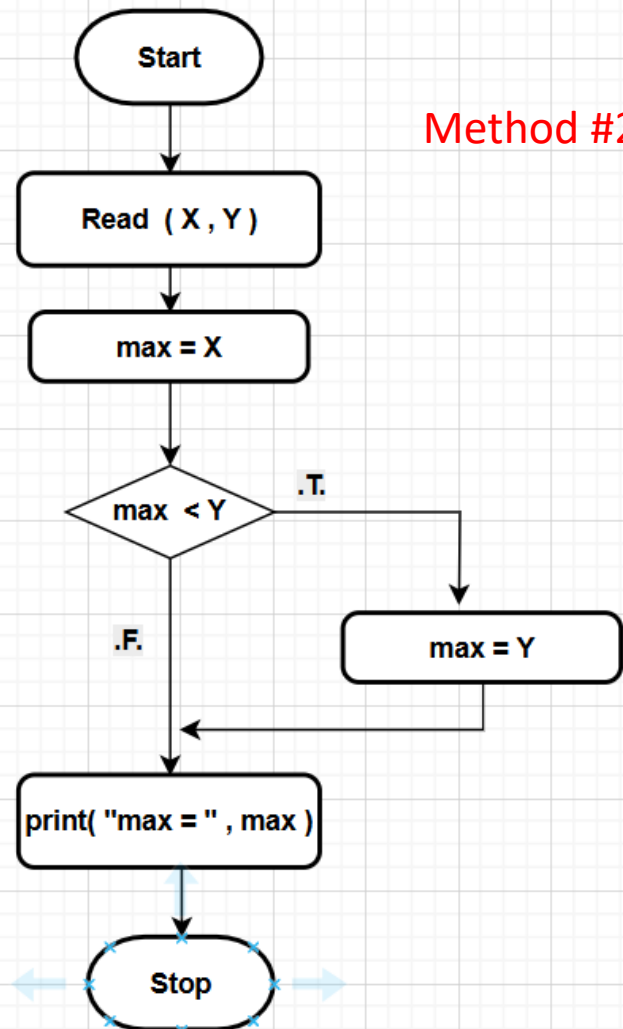


เปรียบเทียบ ?

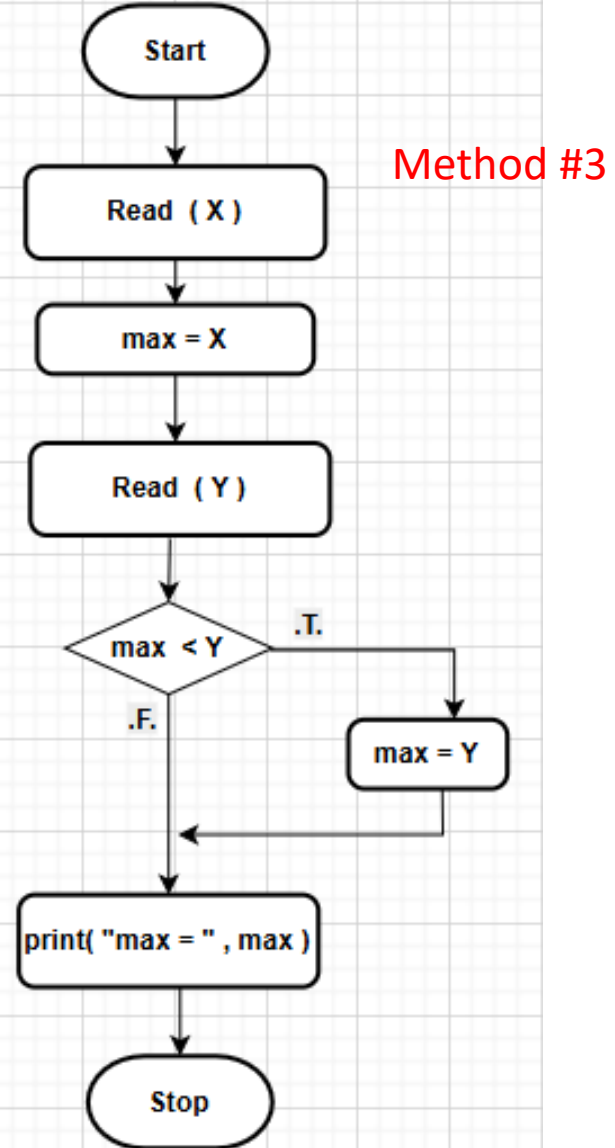
Method #2



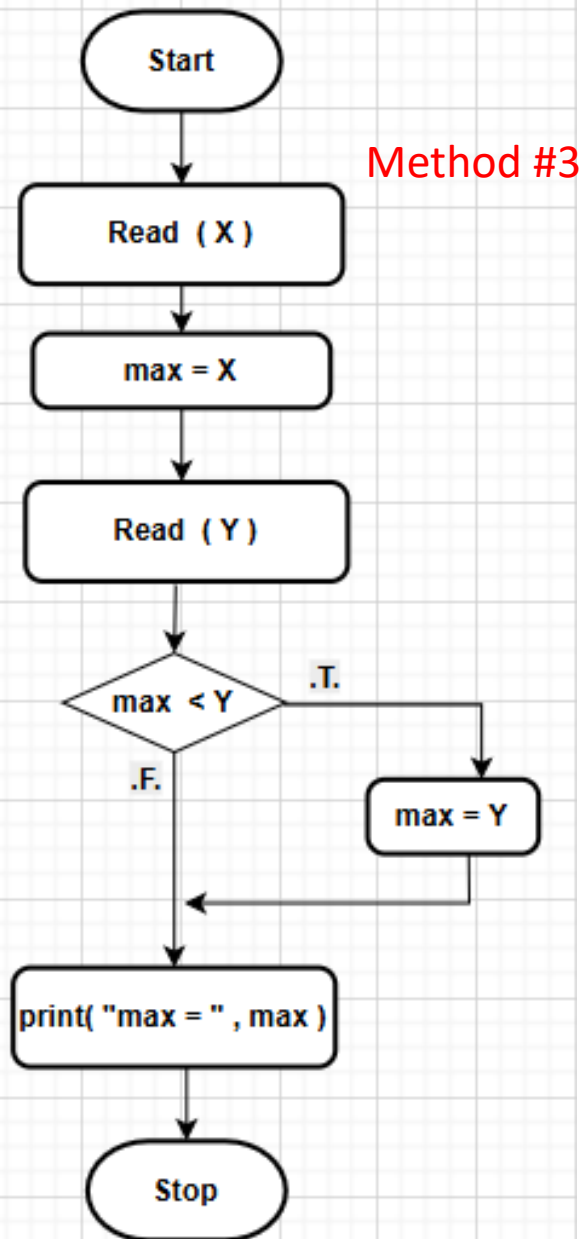
Readability & Writability: หาค่า max 2 ค่า



Rewrite(1)



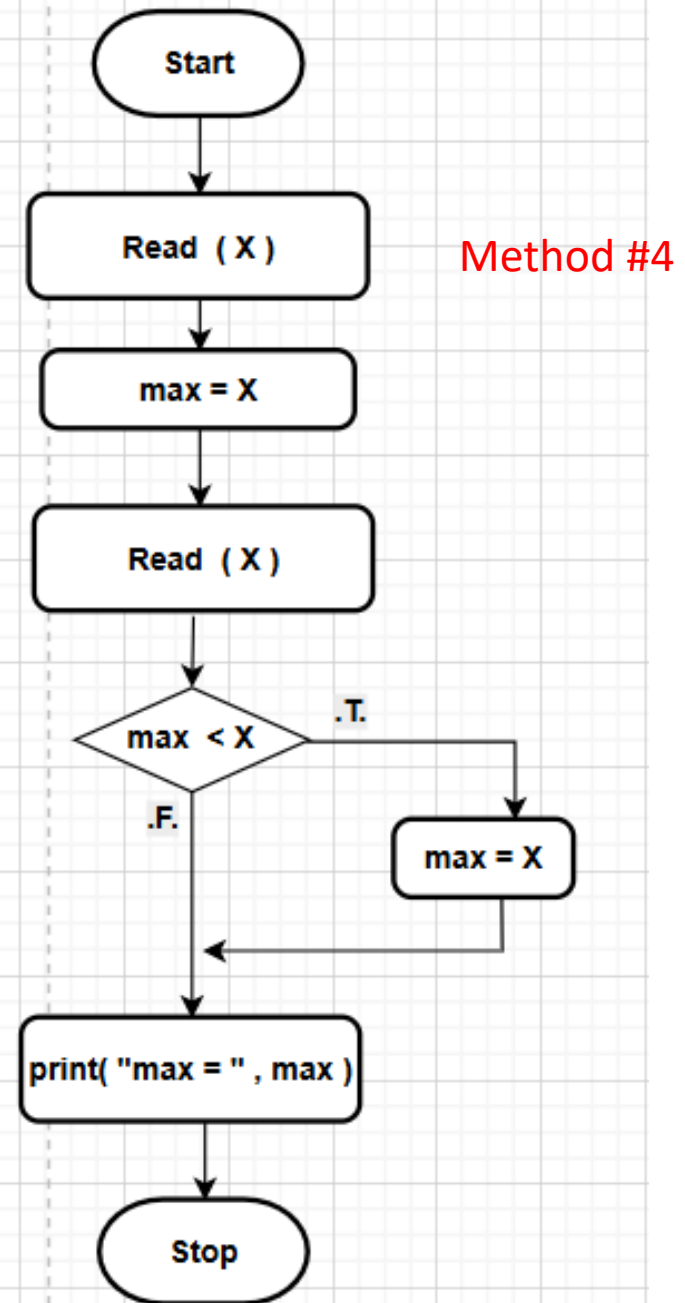
Readability & Writability **หาค่า max 2 ค่า**



Rewrite(2)

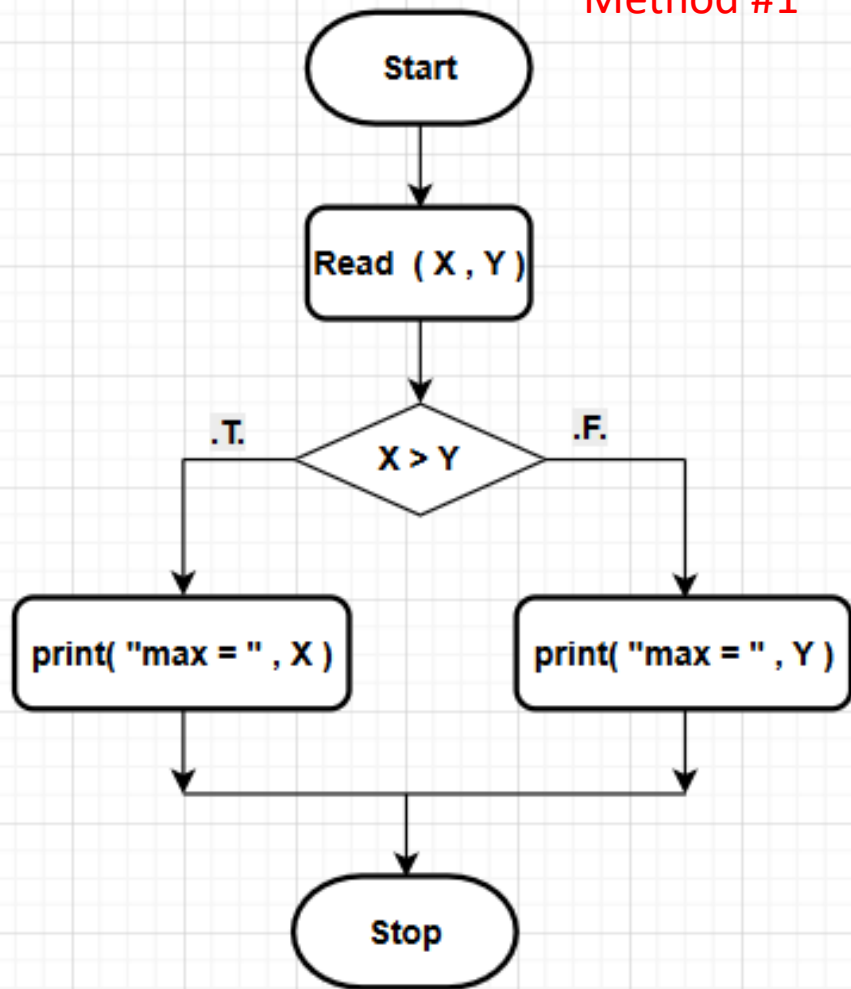


Remove "Y"

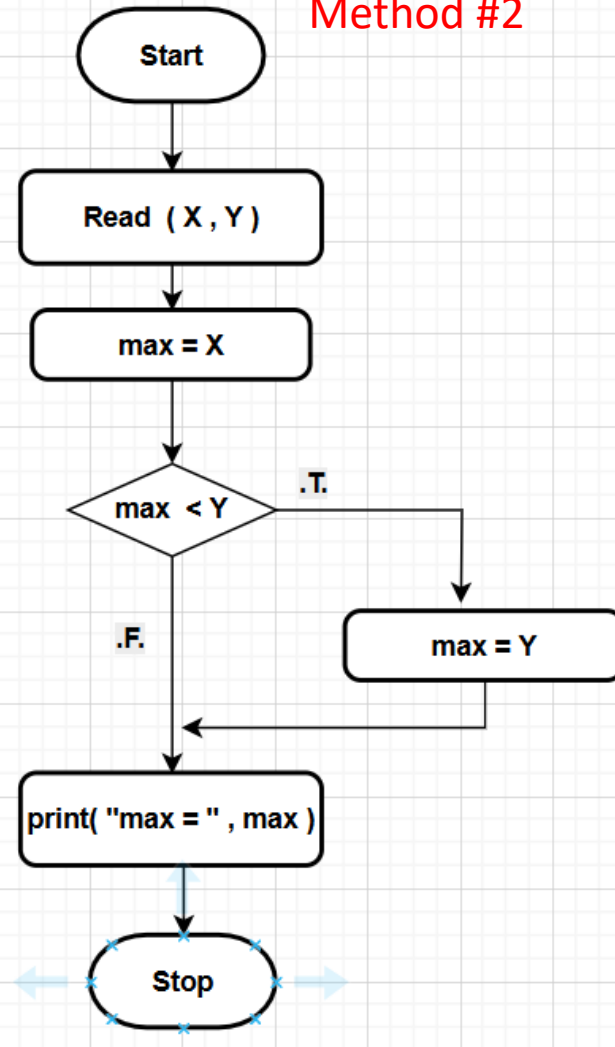


Compare : Readability (หาค่า max 2 ค่า)

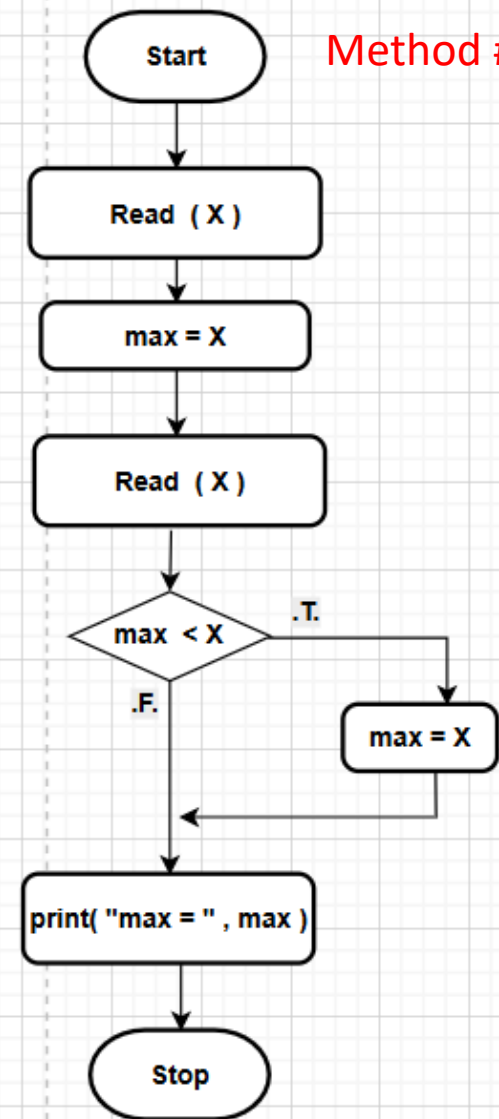
Method #1



Method #2



Method #4



ฝึกเขียน Flow (หาค่า max 3 ค่า)

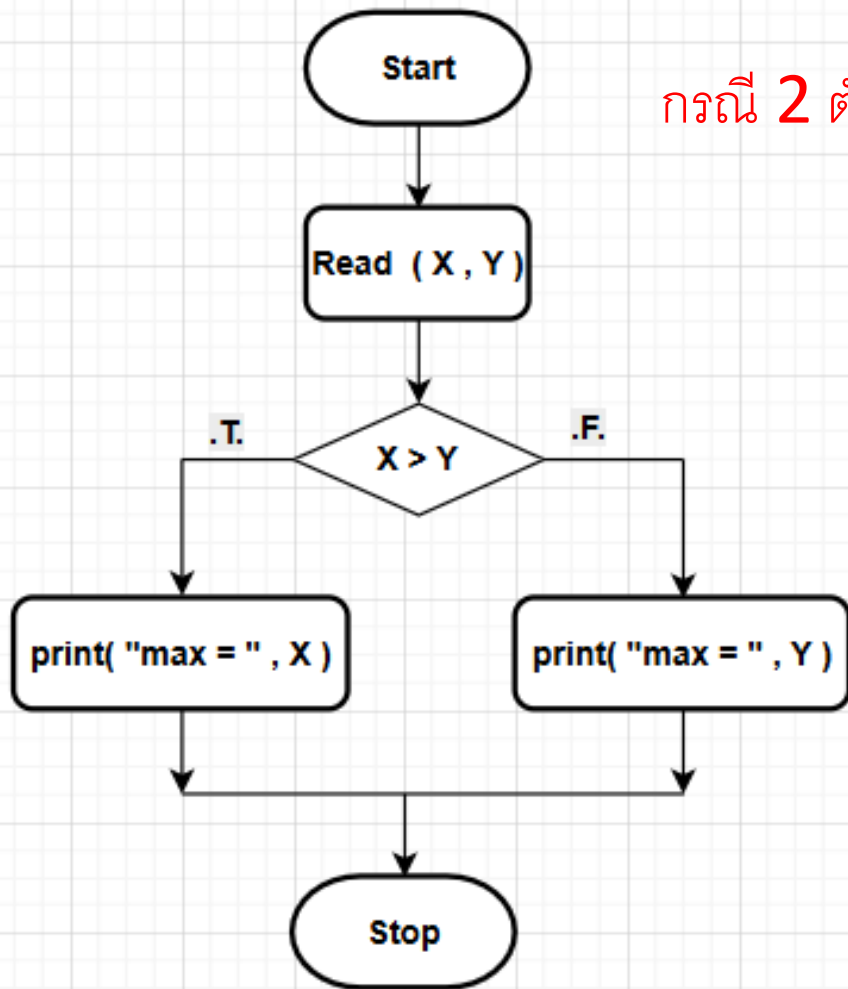
หาค่า **max** จำนวน 3 ค่า โดยให้ **variable Z** เป็นข้อมูลตัวที่ 3

Using method#1 , method#2 and method#4

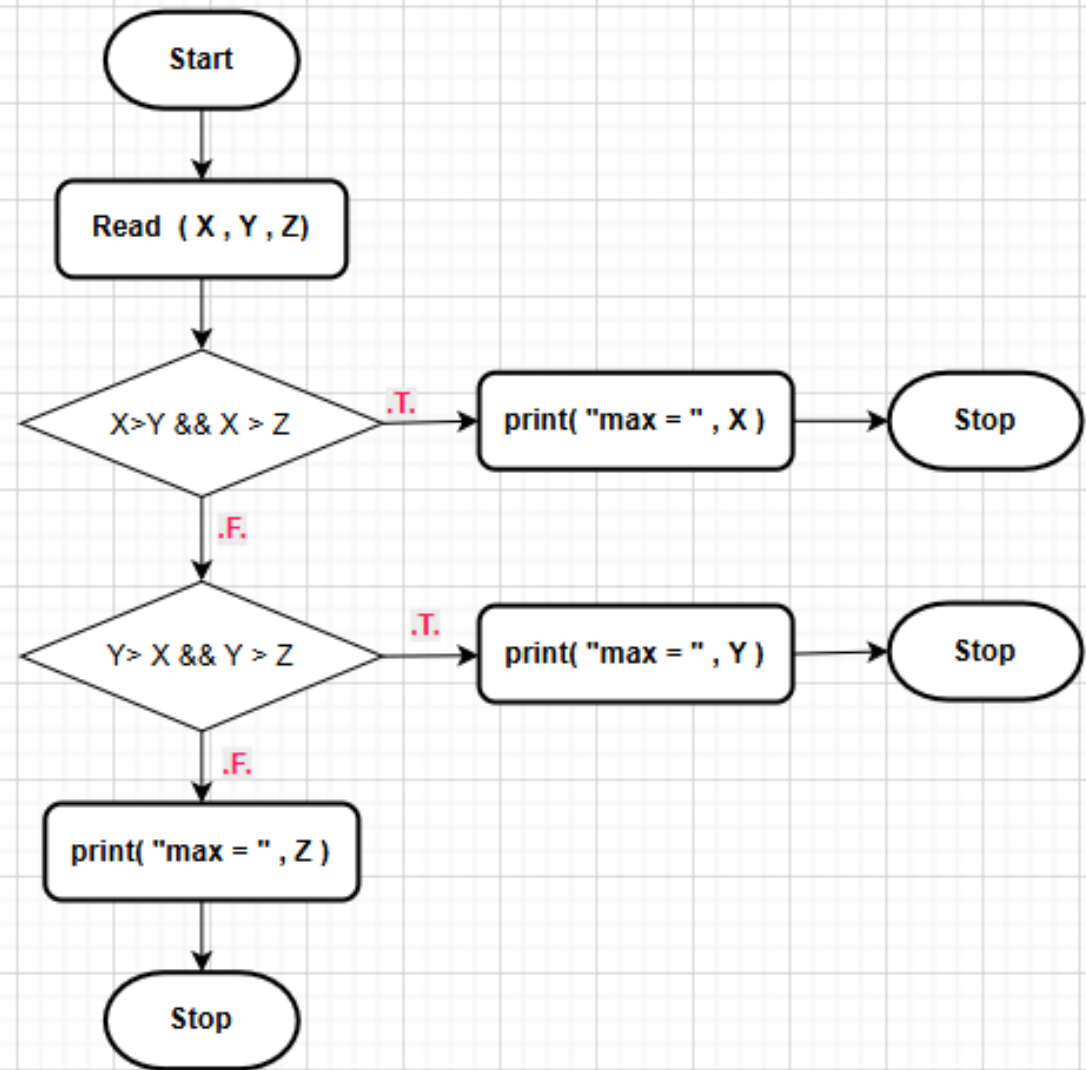
ฝึกเขียน Flow (เลย)

หาค่า max จำนวน 3 ค่า

กรณี 2 ตัวแปร



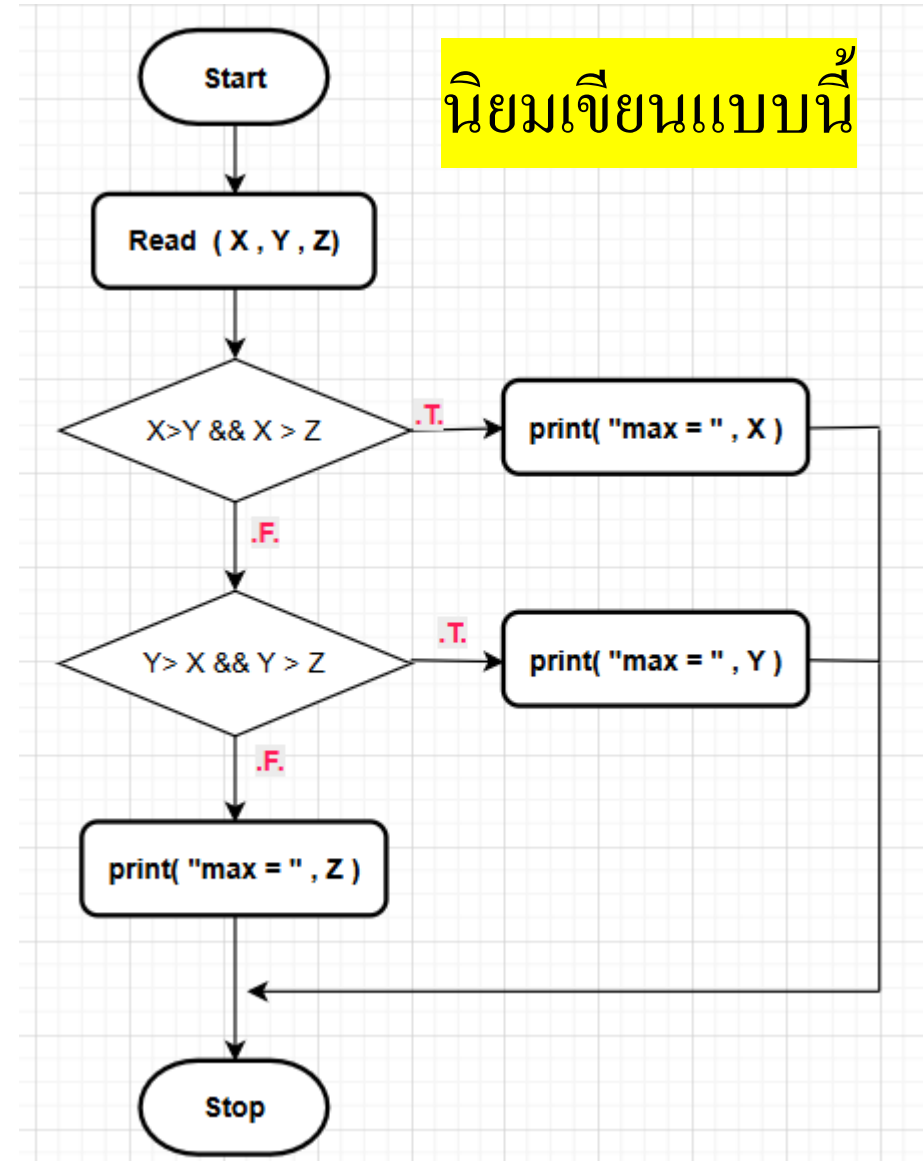
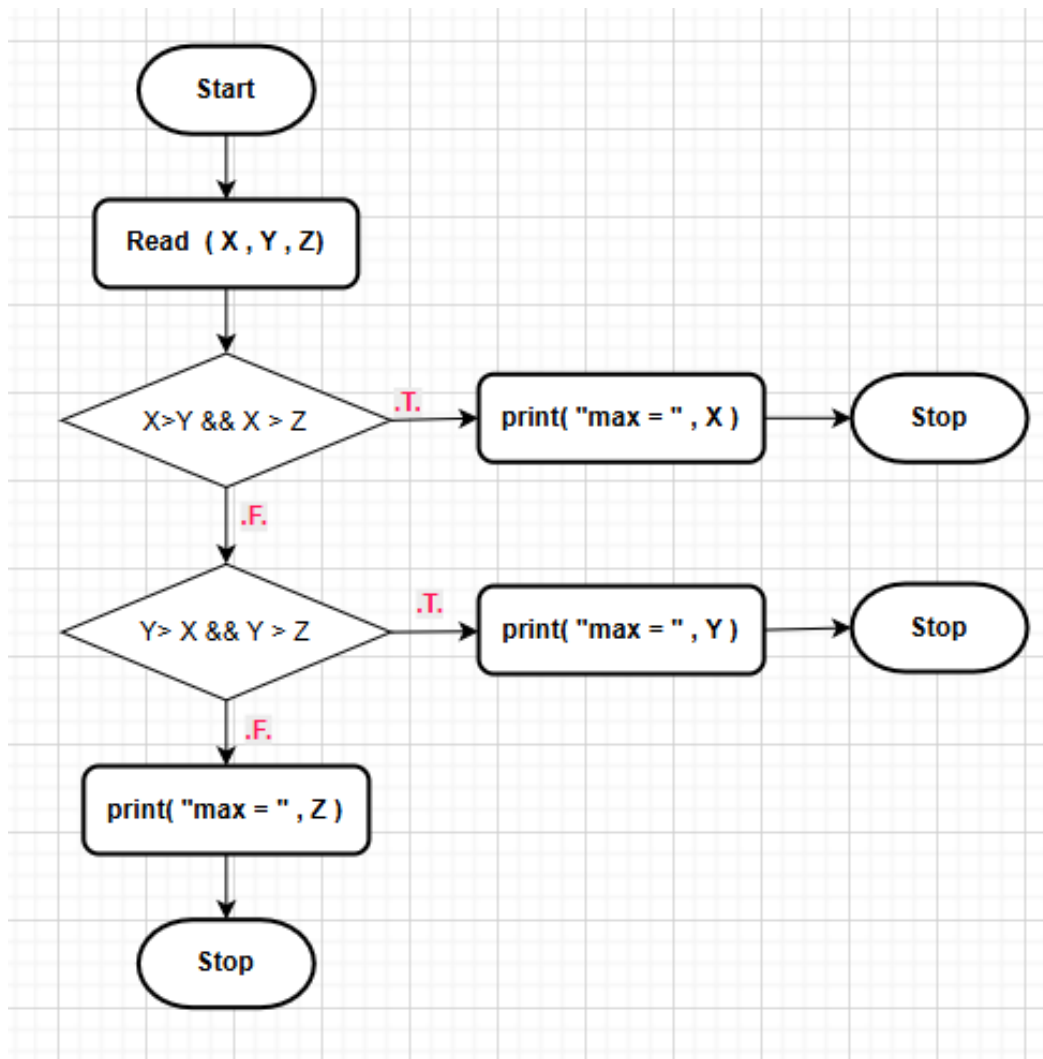
Method #1



ฝึกเขียน Flow (เฉลย)

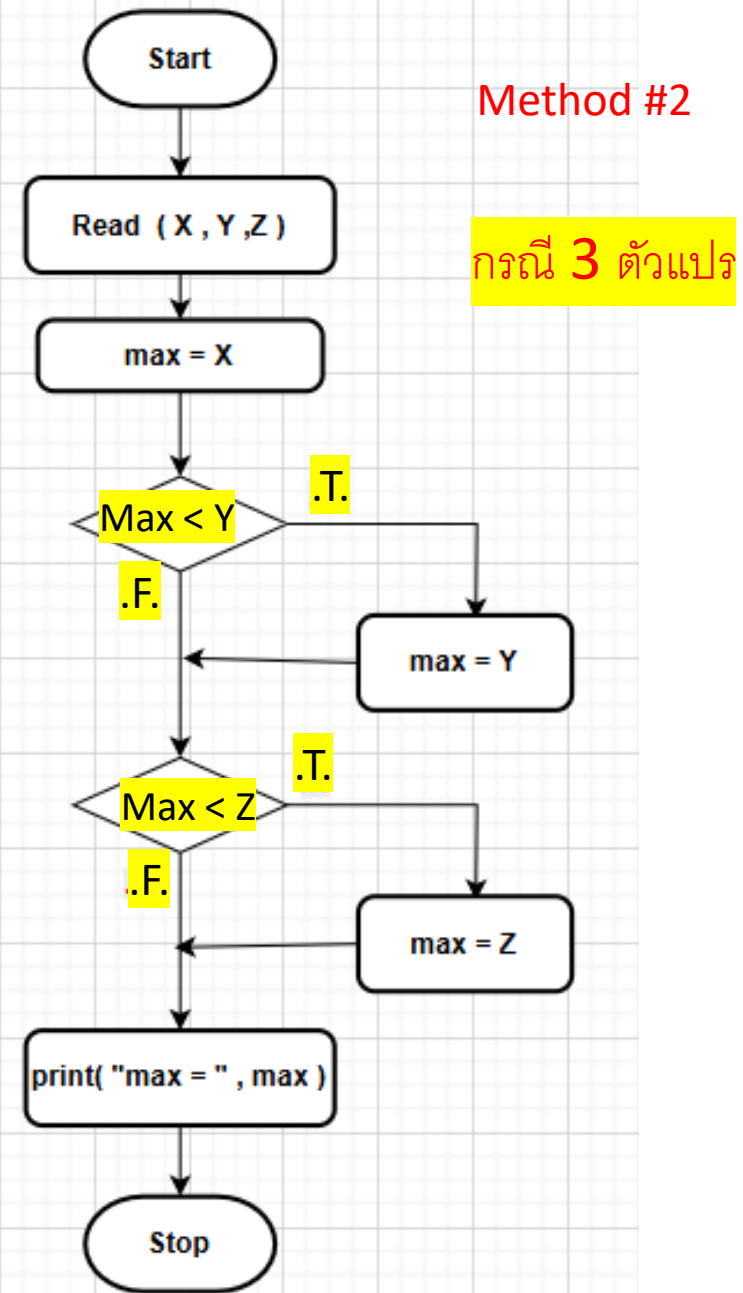
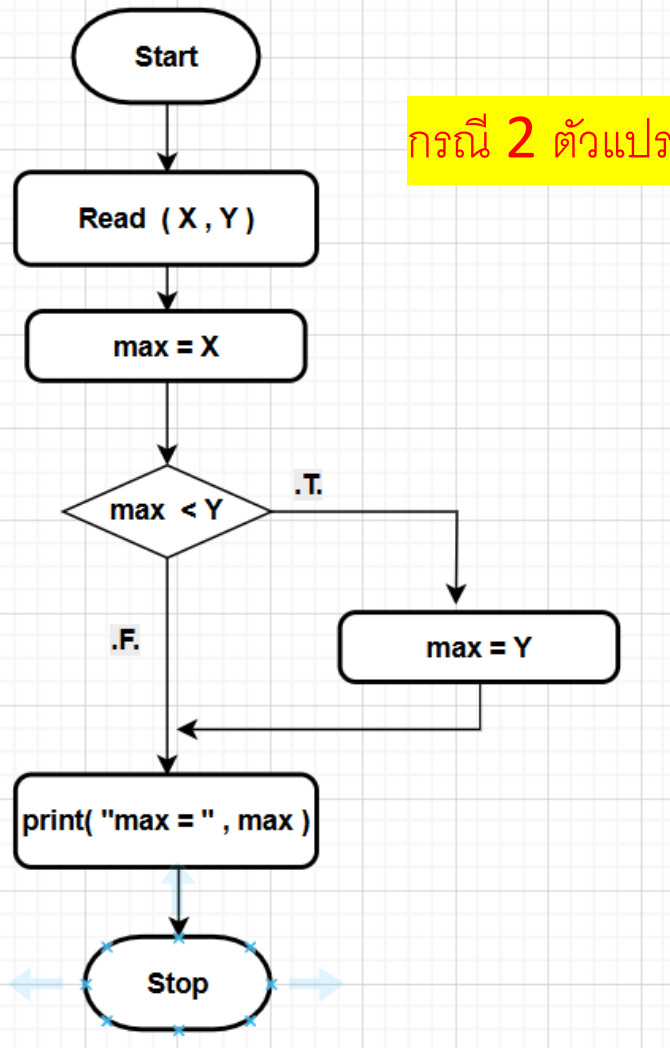
หาค่า max จำนวน 3 ค่า

Method #1



ฝึกเขียน Flow (เฉลย)

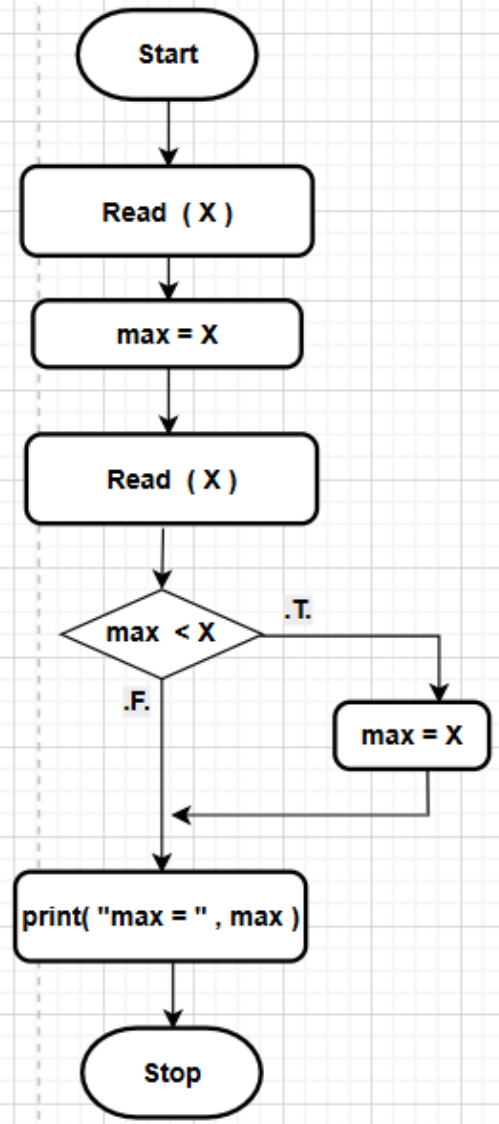
หาค่า max จำนวน 3 ค่า



ฝึกเขียน Flow (เลย)

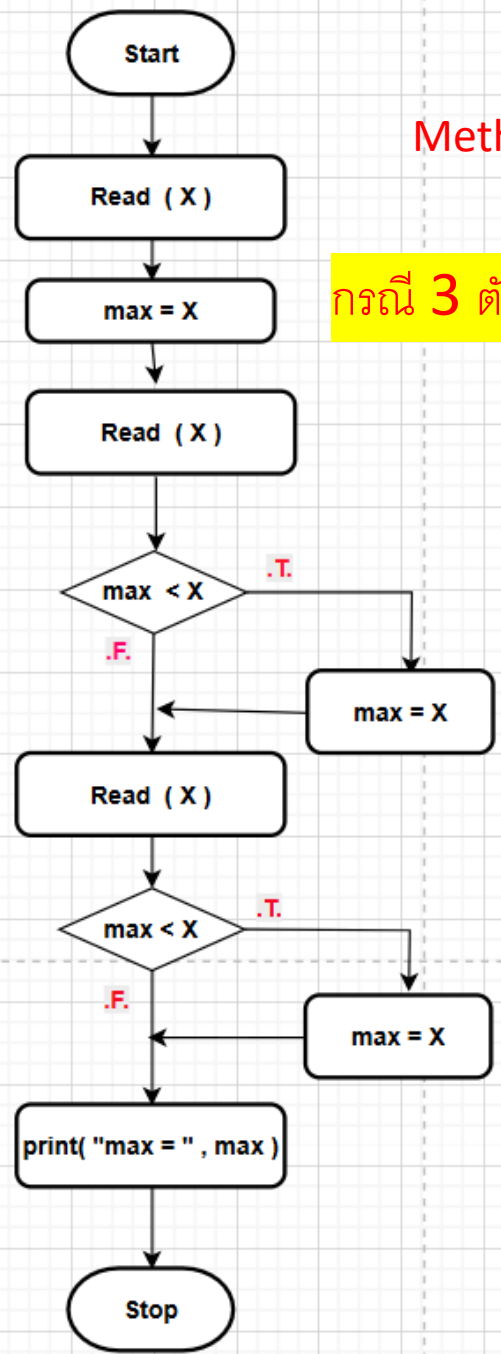
หาค่า max จำนวน 3 ค่า

กรณี 2 ตัวแปร



Method #4

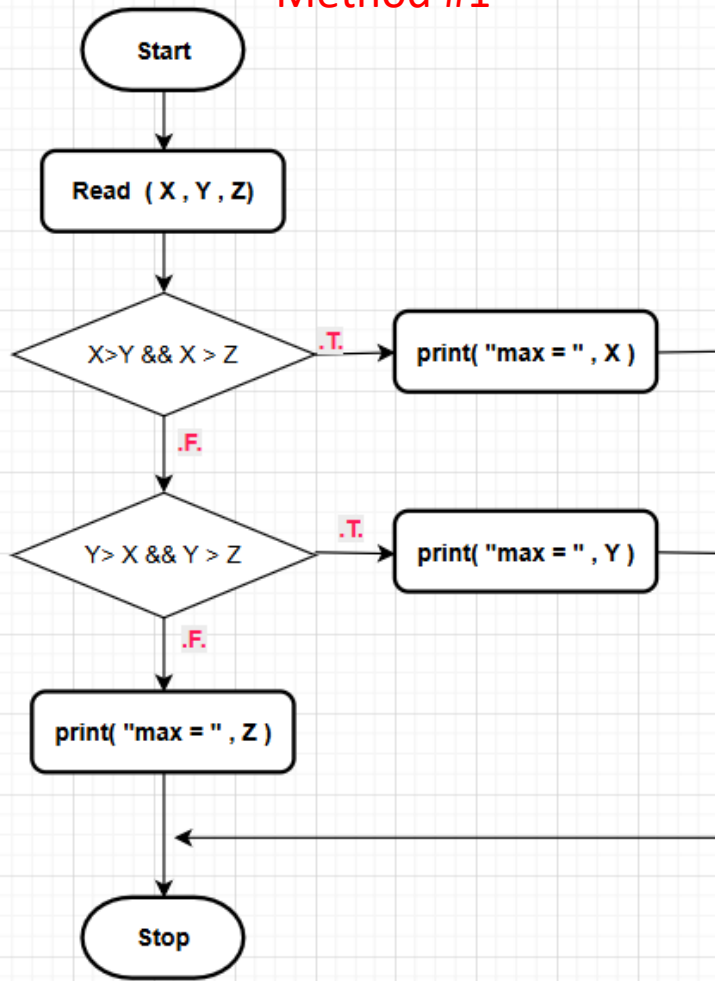
กรณี 3 ตัวแปร



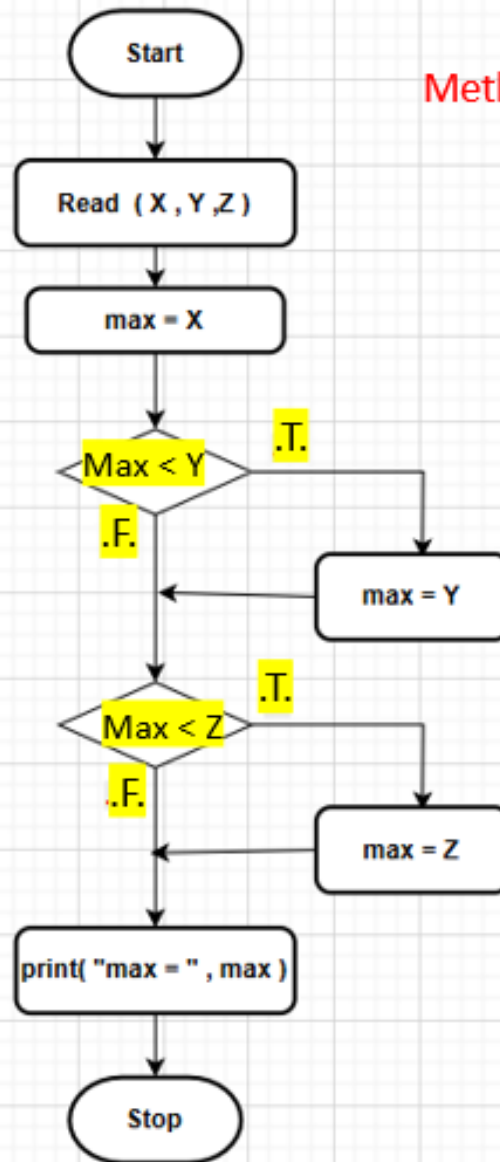
Compare Readability & writability

หาค่า **max** จำนวน 3 ค่า

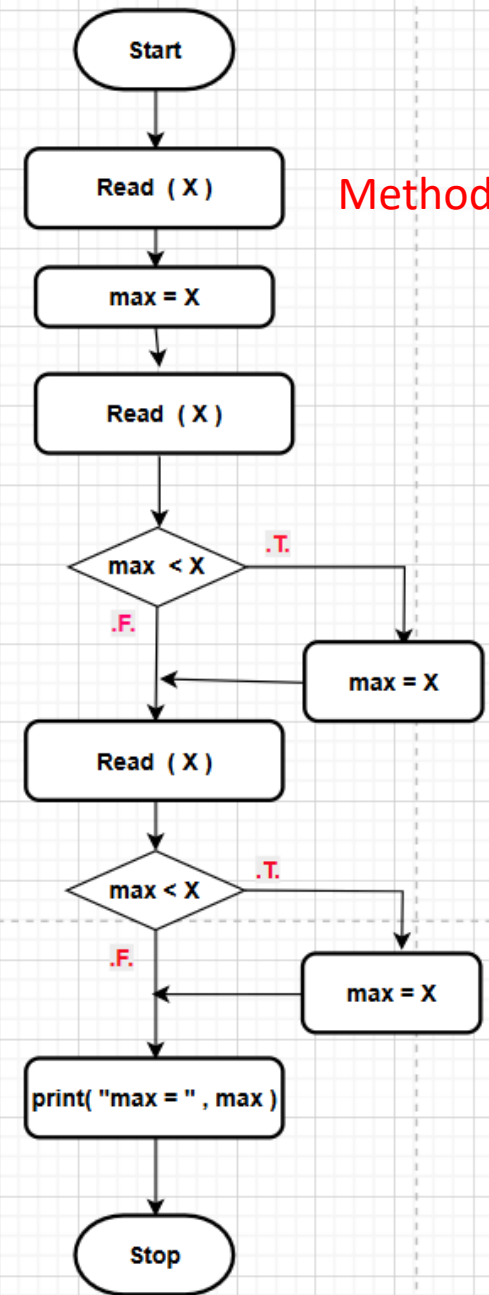
Method #1



Method #2



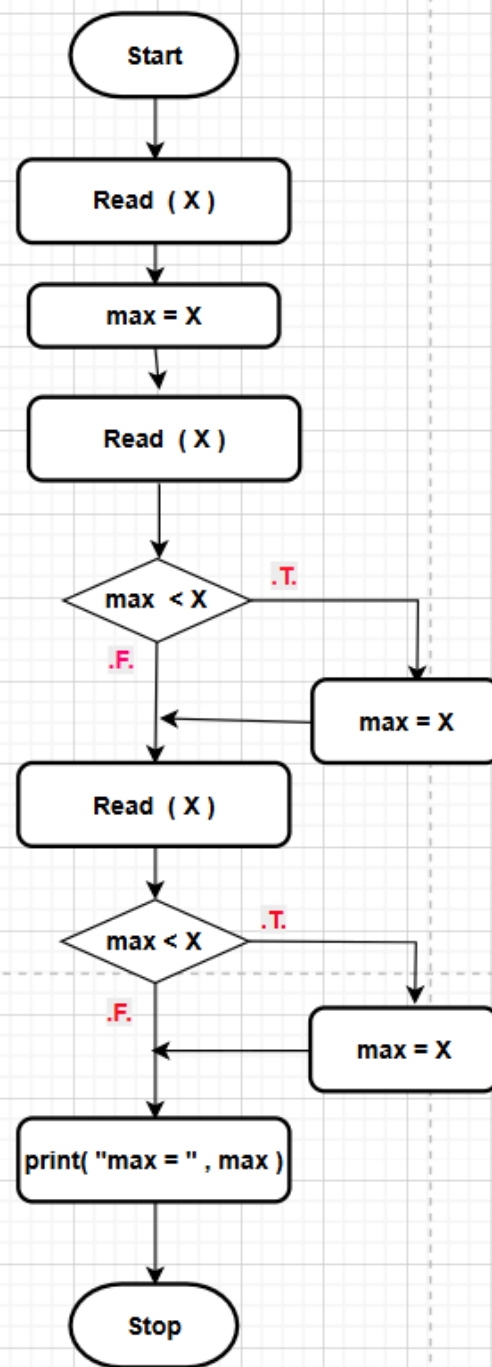
Method #4



ฝึกเขียน Flow

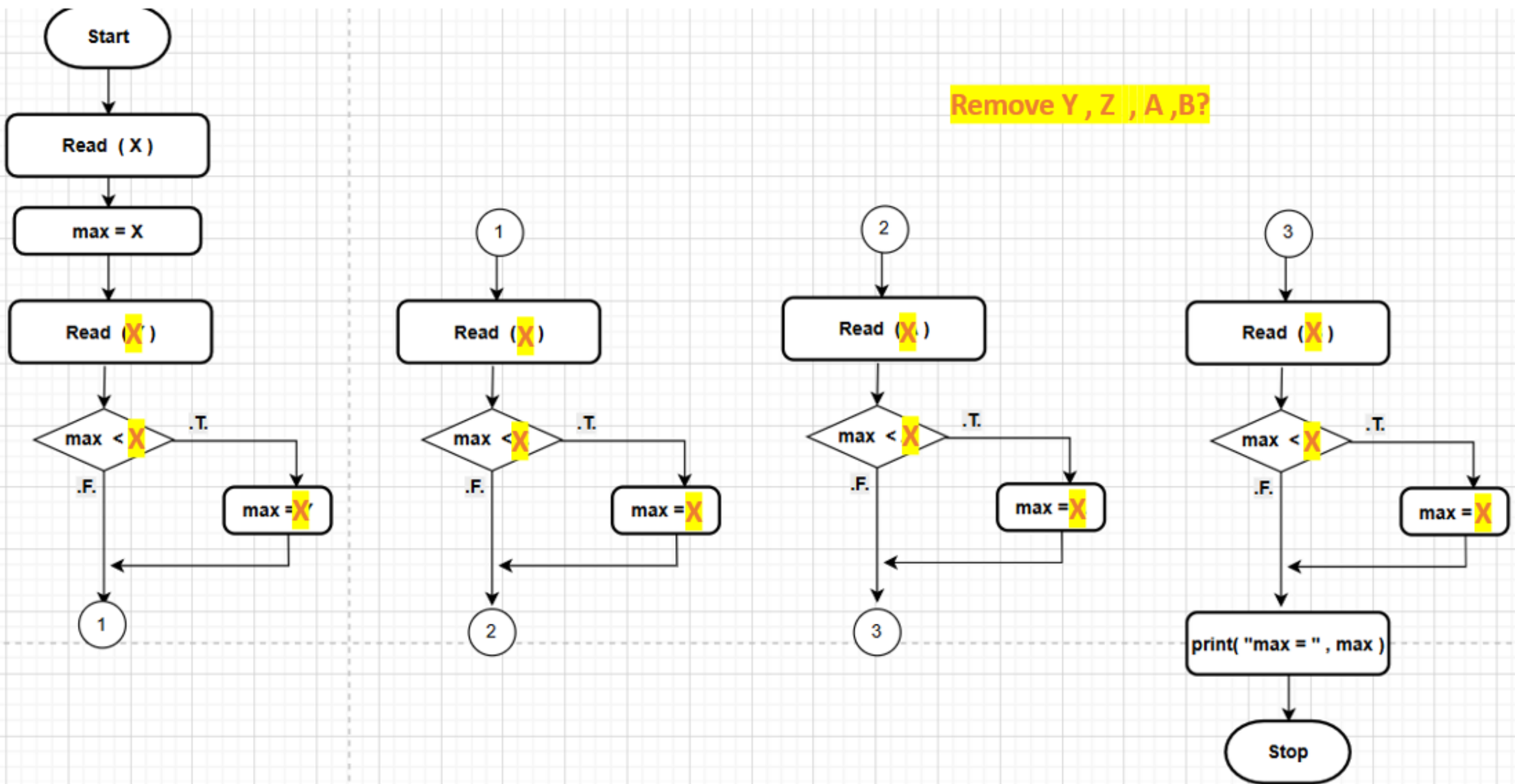
หาค่า max จำนวน 5 ค่า

แปลง flow ให้เป็นหาค่า max 5 ค่า



ฝึกเขียน Flow

หาค่า max จำนวน 5 ค่า

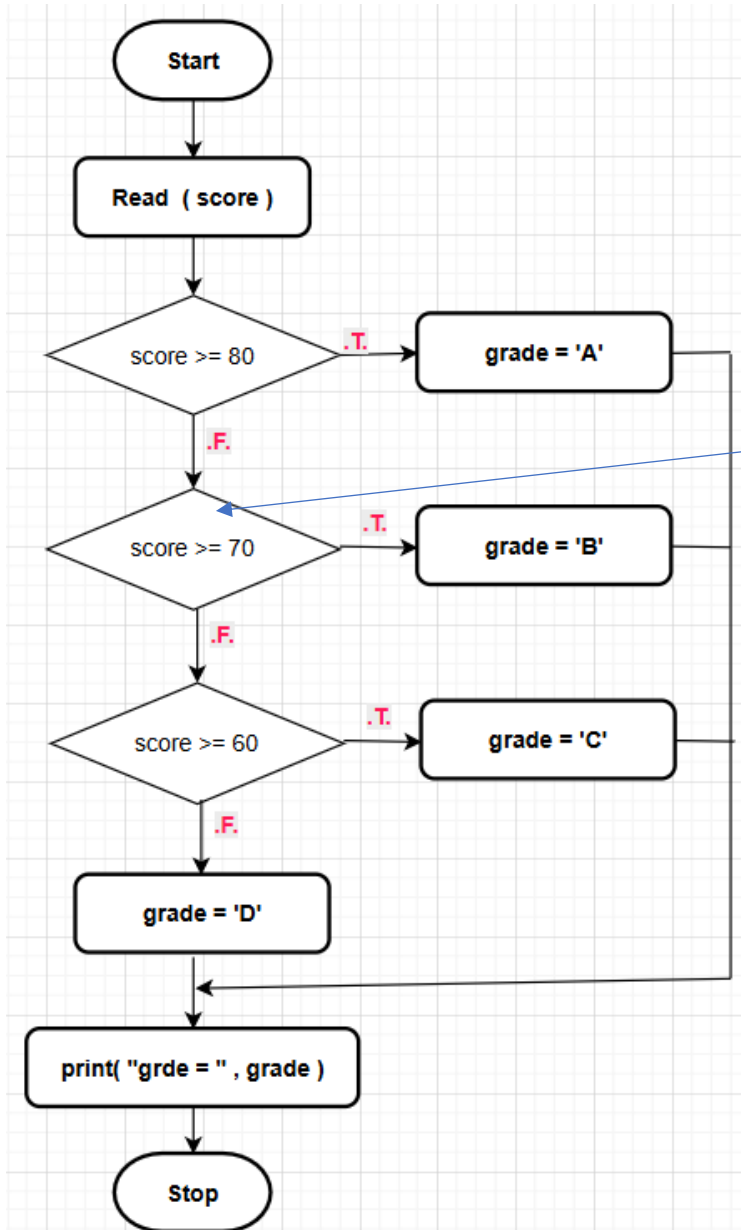


จงเขียน โปรแกรมคำนวณค่าเกรด

เขียนโปรแกรมรับคะแนนสอบ และแสดงผลว่าเป็นเกรดอะไร ตามเงื่อนไขต่อไปนี้:

- ถ้าคะแนน ≥ 80 ให้แสดง "เกรด A"
- ถ้าคะแนน ≥ 70 แต่ < 80 ให้แสดง "เกรด B"
- ถ้าคะแนน ≥ 60 แต่ < 70 ให้แสดง "เกรด C"
- ถ้าต่ำกว่า 60 ให้แสดง "D"

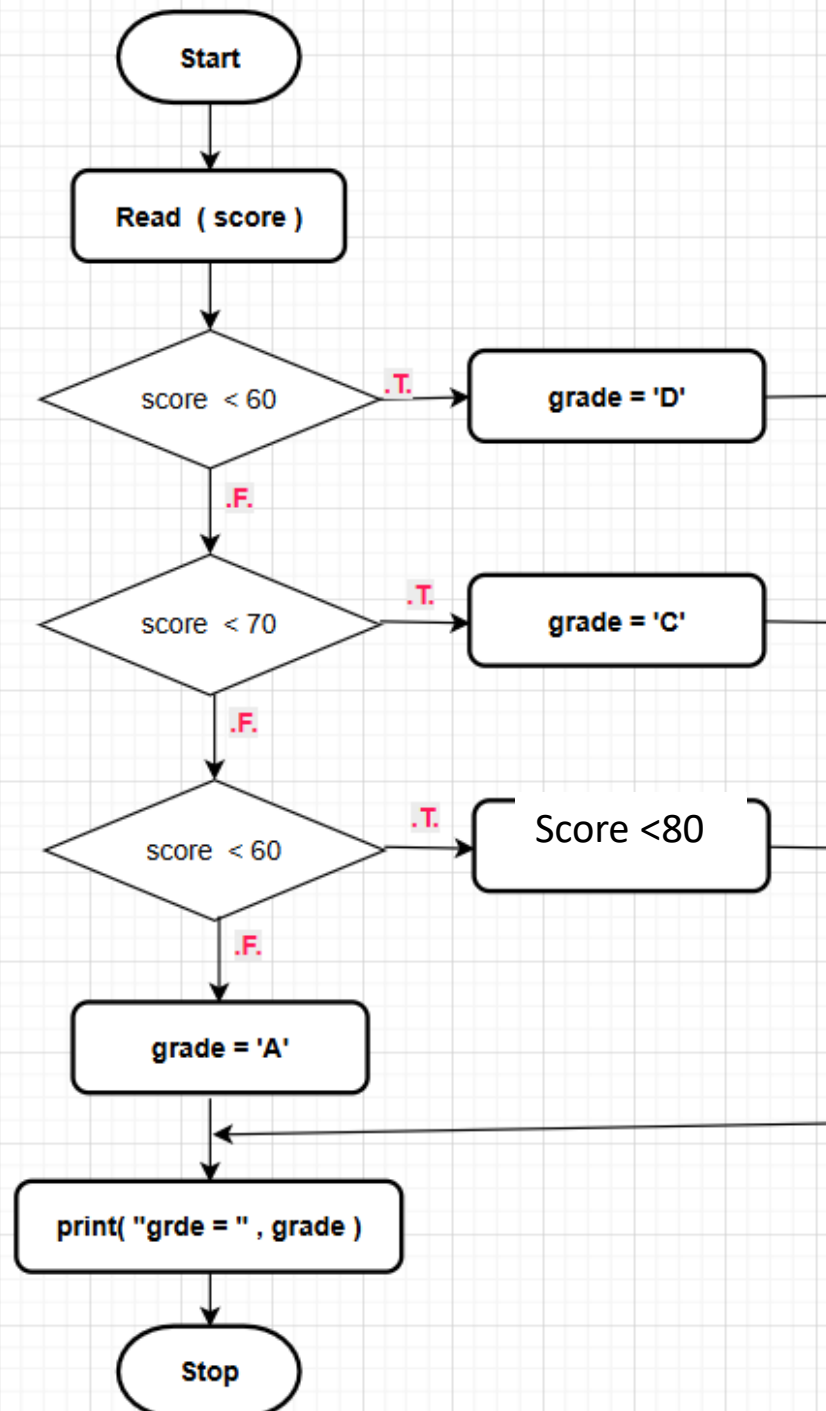
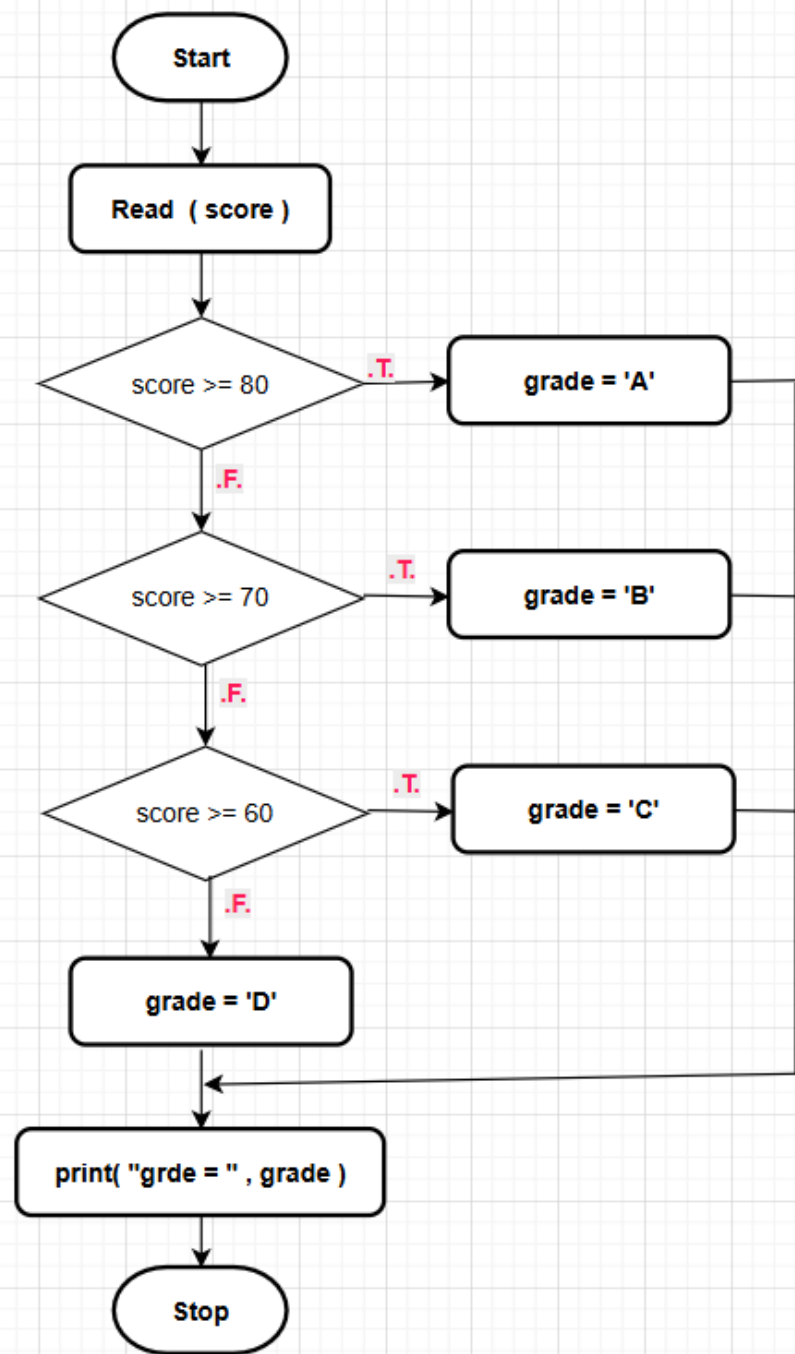
จงเขียน โปรแกรมคำนวณค่าเกรด



เขียนโปรแกรมรับคะแนนสอบ และแสดงผลว่าเป็นเกรดอะไร ตามเงื่อนไขต่อไปนี้:

- ถ้าคะแนน ≥ 80 ให้แสดง "เกรด A"
- ถ้าคะแนน ≥ 70 แต่ < 80 ให้แสดง "เกรด B"
- ถ้าคะแนน ≥ 60 แต่ < 70 ให้แสดง "เกรด C"
- ถ้าต่ำกว่า 60 ให้แสดง "D"

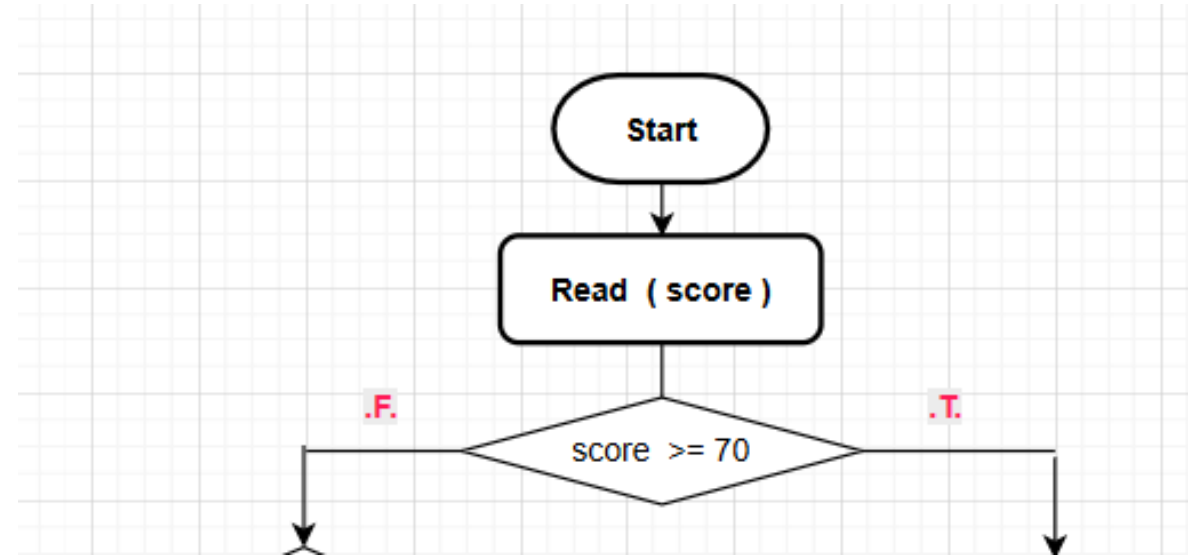
ทำไมไม่ใช่ `score >= 70 && score < 80` ?



เขียนโปรแกรมรับคะแนนสอบ และแสดงผลว่าเป็นเกรดอะไร ตามเงื่อนไขต่อไปนี้:

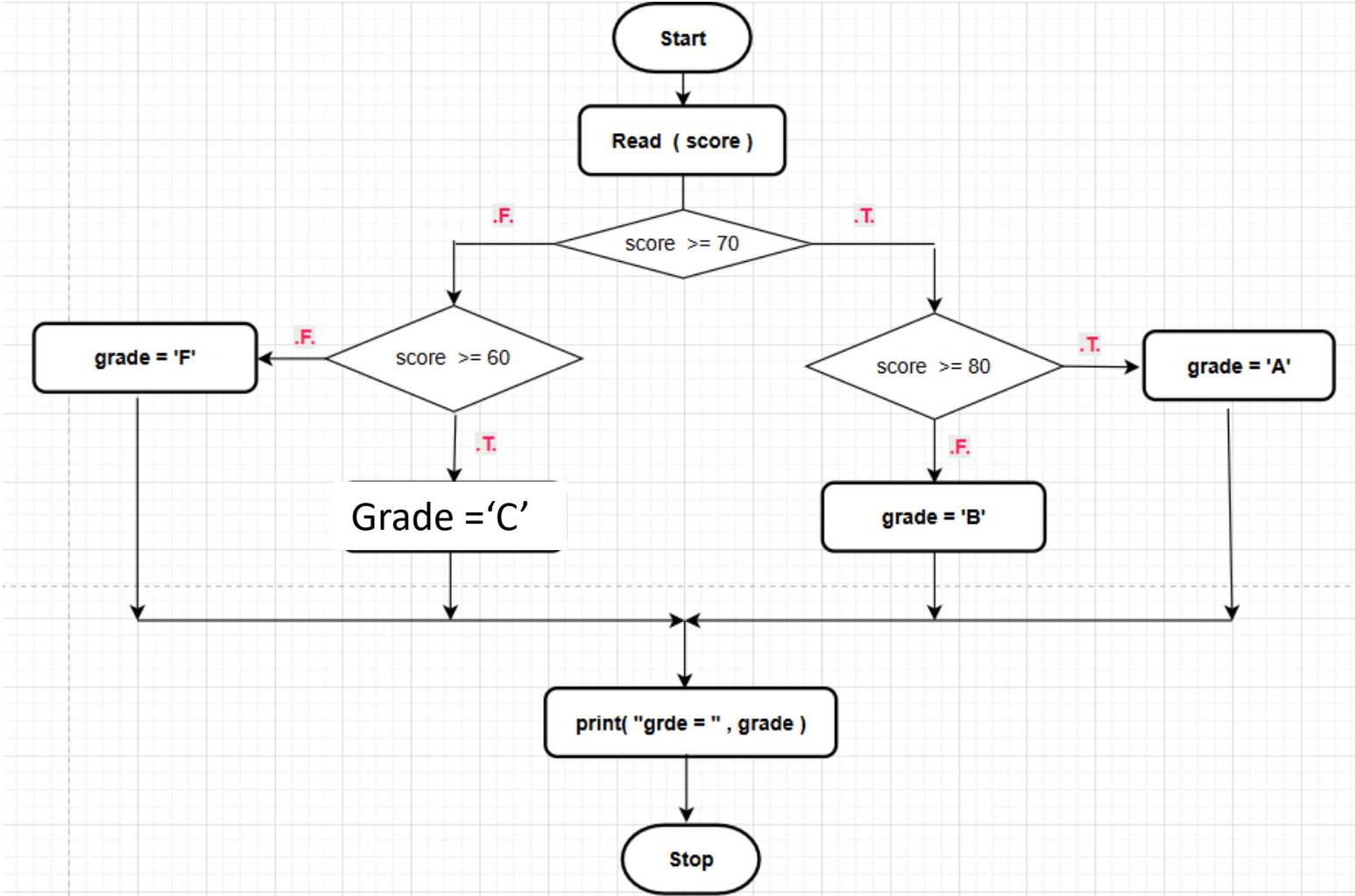
- ถ้าคะแนน ≥ 80 ให้แสดง "เกรด A"
- ถ้าคะแนน ≥ 70 แต่ < 80 ให้แสดง "เกรด B"
- ถ้าคะแนน ≥ 60 แต่ < 70 ให้แสดง "เกรด C"
- ถ้าต่ำกว่า 60 ให้แสดง "D"

ให้ใส่ condition flow?



เขียนโปรแกรมรับคะแนนสอบ และแสดงผลว่าเป็นเกรดอะไร ตามเงื่อนไขต่อไปนี้:

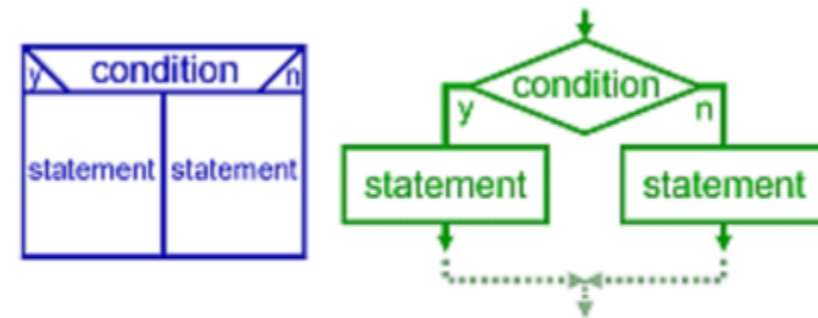
- ถ้าคะแนน ≥ 80 ให้แสดง "เกรด A"
- ถ้าคะแนน ≥ 70 แต่ < 80 ให้แสดง "เกรด B"
- ถ้าคะแนน ≥ 60 แต่ < 70 ให้แสดง "เกรด C"
- ถ้าต่ำกว่า 60 ให้แสดง "D"



Selection Structure in Pascal

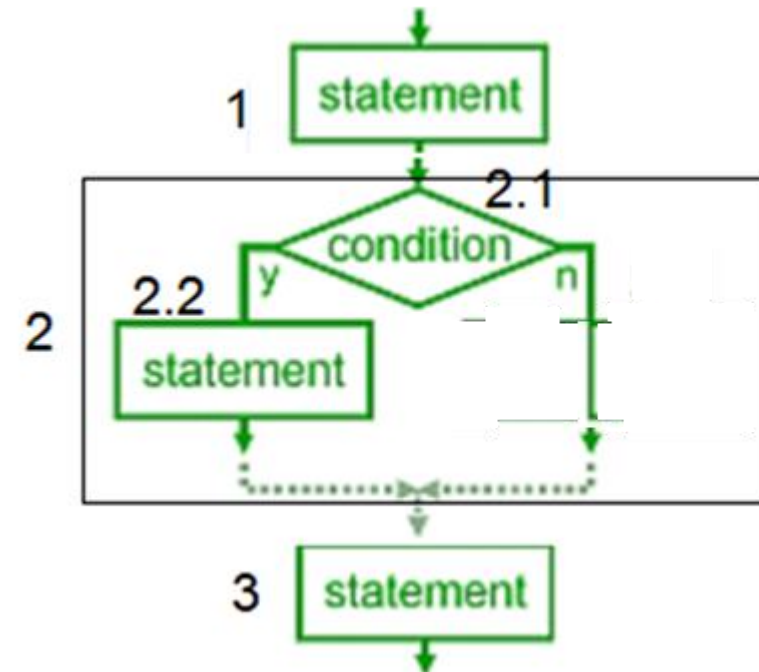
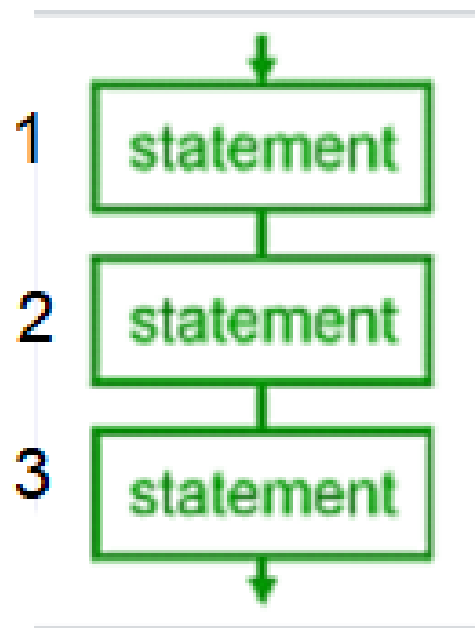
There are three types :

- ❑ **if-then-statement (one-way)**
- ❑ **if-then-else statement (two-way)**
- ❑ **Ladder- If-then-else**



If-then-else

if - then statement (one way selection)



Execution flow

Condition is true

```
5  if x > 0 then
6  begin
7  → writeln('x เป็นจำนวนบวก');
8     writeln('โปรแกรมทำงานในส่วน if');
9  end;
10 writeln('code after if');
11
```

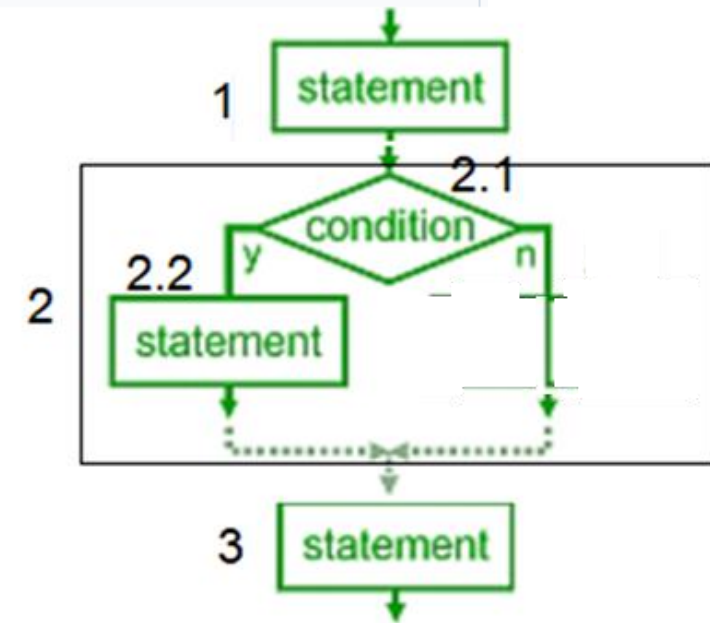
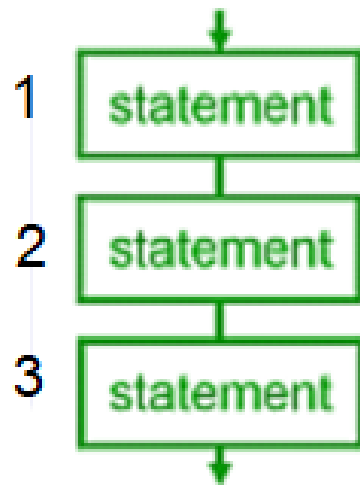
// code after if

Condition is false

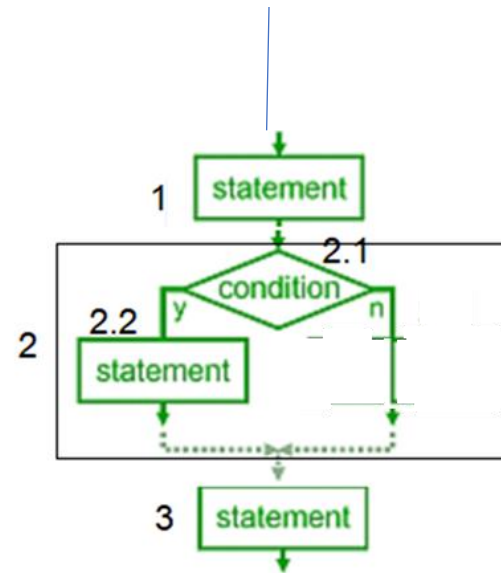
```
int number = 5;
```

```
5  if x > 0 then
6  begin
7     writeln('x เป็นจำนวนบวก');
8     writeln('โปรแกรมทำงานในส่วน if');
9  end;
10 writeln('code after if');
11
```

How if Statement Works



```
if condition then
begin
    statement1;
    statement2;
end;
```



```
if condition then
    statement;
```

```
4
5 if x > 0 then
6 begin
7     writeln('x เป็นจำนวนบวก');
8     writeln('โปรแกรมทำงานในส่วน if');
9 end;
10
```

```
4
5 if x > 0 then
6 |     writeln('x is positive');
7
```

```

1
2 program PopulationCheck;
3 var
4     population: integer;
5
6 begin
7     Write('How many the population: ');
8     ReadLn(population);
9
10    if population >= 185086 then
11        WriteLn('the 100 largest U.S. city');
12
13 end.
14

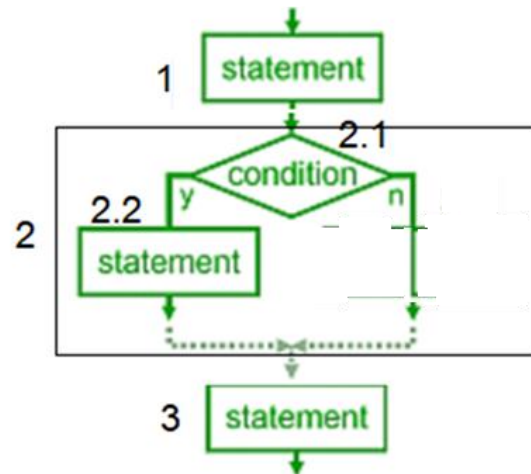
```

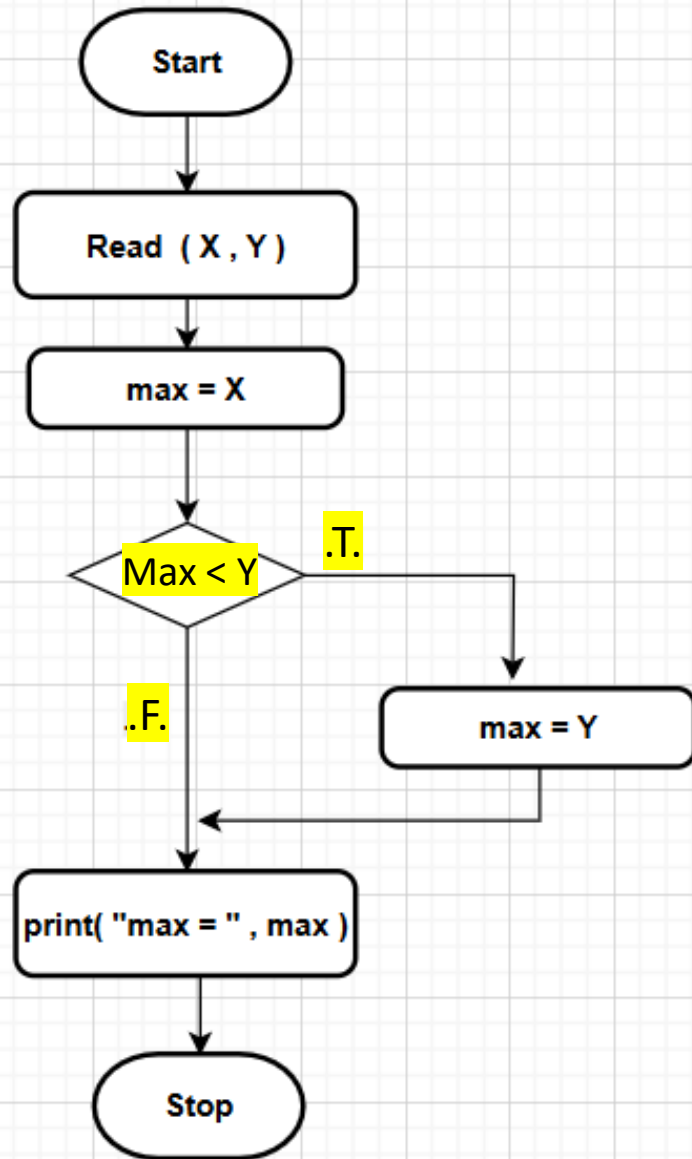
```

1
2 program PopulationCheck;
3 var
4     population: integer;
5
6 begin
7     Write('How many the population: ');
8     ReadLn(population);
9
10    if population >= 185086 then
11        begin
12            WriteLn('the 100 largest U.S. city');
13        end;
14
15 end.
16

```

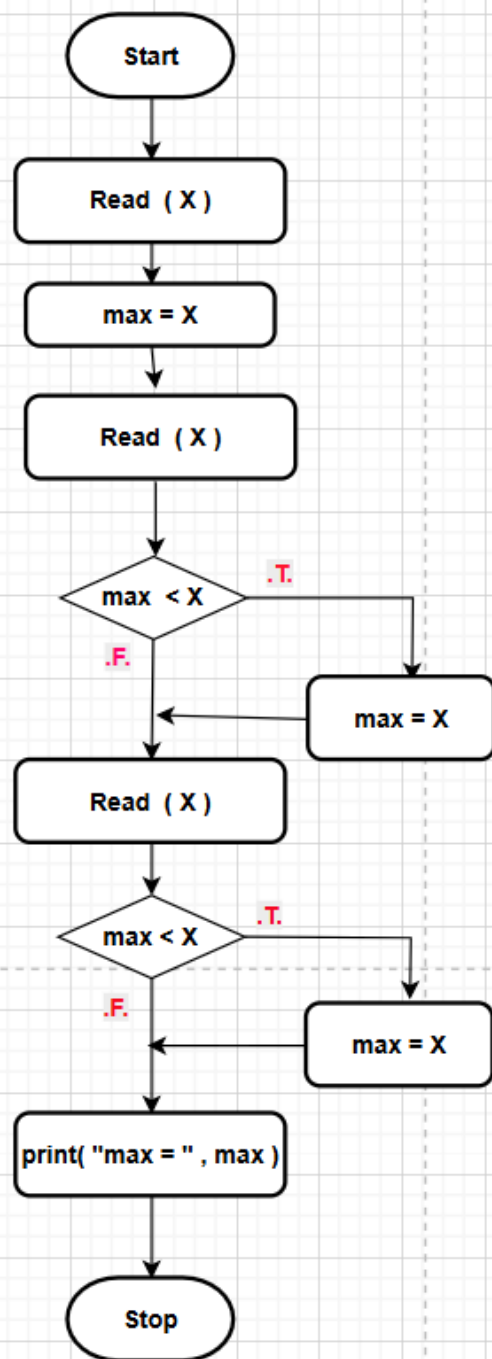
PopulationCheck.pas





Convert to Pascal

```
program maxM1;  
var  
  
begin  
  
end.
```

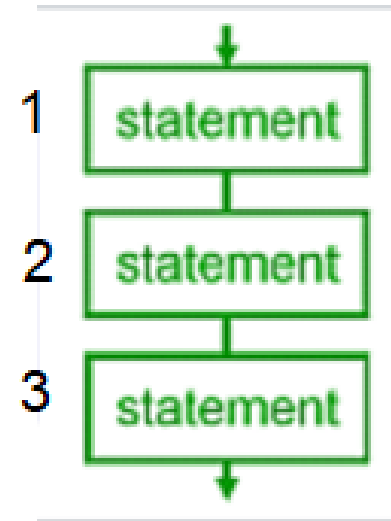
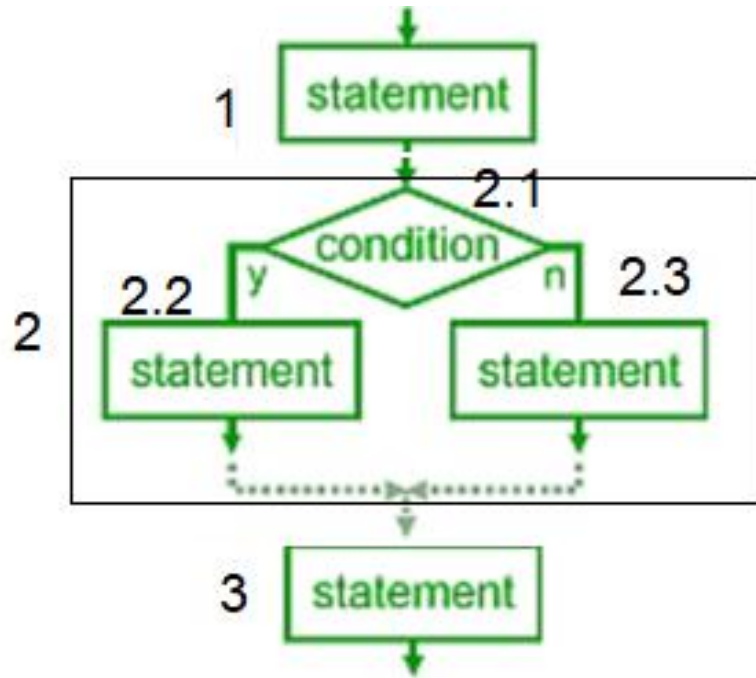


Convert to PASCAL

```
program maxM2;  
var  
  
begin  
  
end.
```

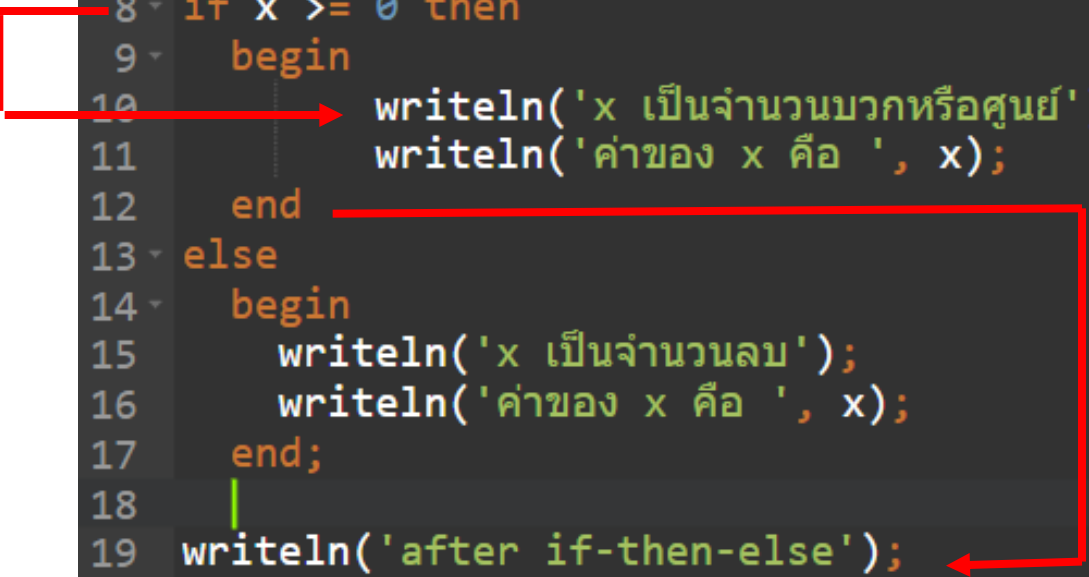

2. Selection Structure

if -then -else statement
(two-way structure)



Condition is true

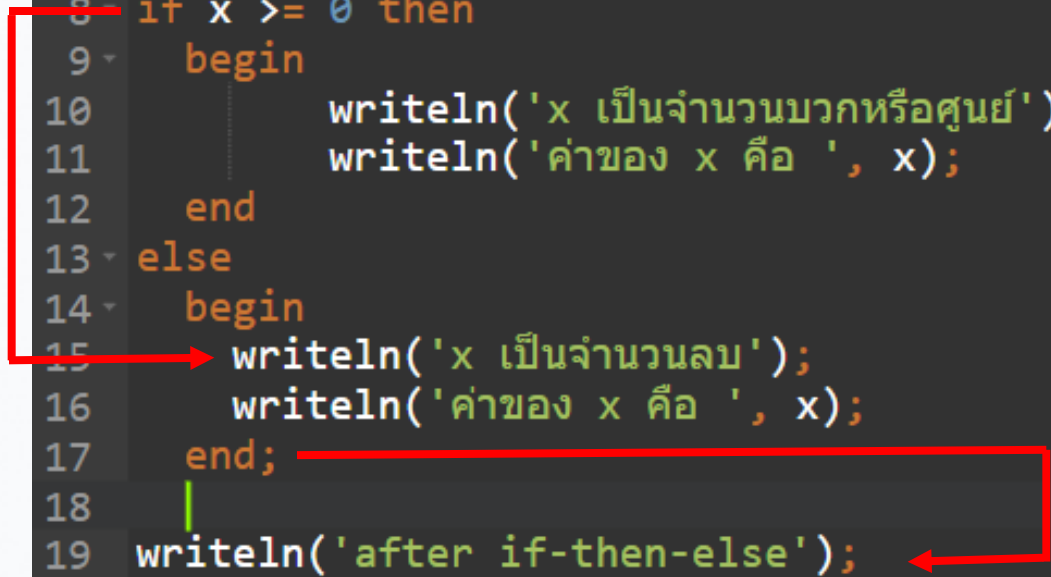
```
7
8 if x >= 0 then
9   begin
10    writeln('x เป็นจำนวนบวกหรือศูนย์');
11    writeln('ค่าของ x คือ ', x);
12   end
13 else
14   begin
15    writeln('x เป็นจำนวนลบ');
16    writeln('ค่าของ x คือ ', x);
17   end;
18
19 writeln('after if-then-else');
20
```



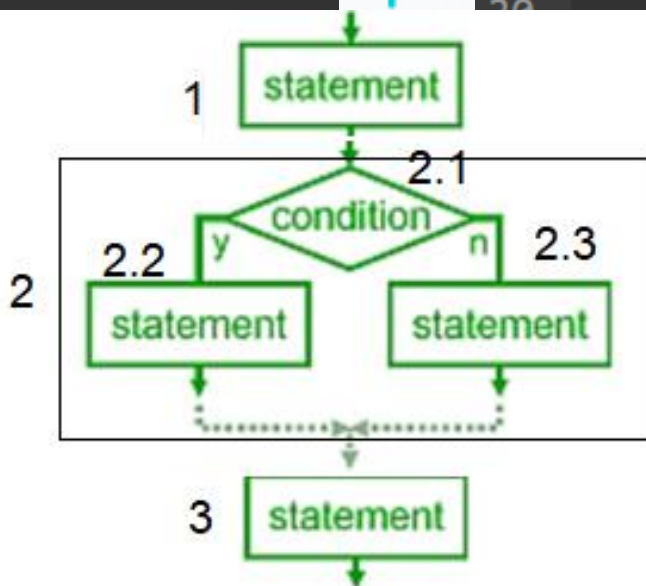
// code after if...

Condition is false

```
7
8 if x >= 0 then
9   begin
10    writeln('x เป็นจำนวนบวกหรือศูนย์');
11    writeln('ค่าของ x คือ ', x);
12   end
13 else
14   begin
15    writeln('x เป็นจำนวนลบ');
16    writeln('ค่าของ x คือ ', x);
17   end;
18
19 writeln('after if-then-else');
20
```



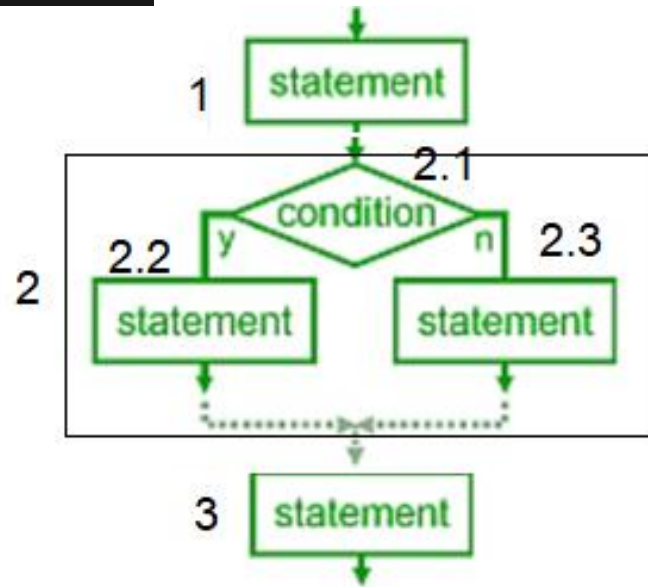
// code after if...else



```

if condition then
begin
    statement1;
    statement2;
end
else
begin
    statement3;
    statement4;
end;

```



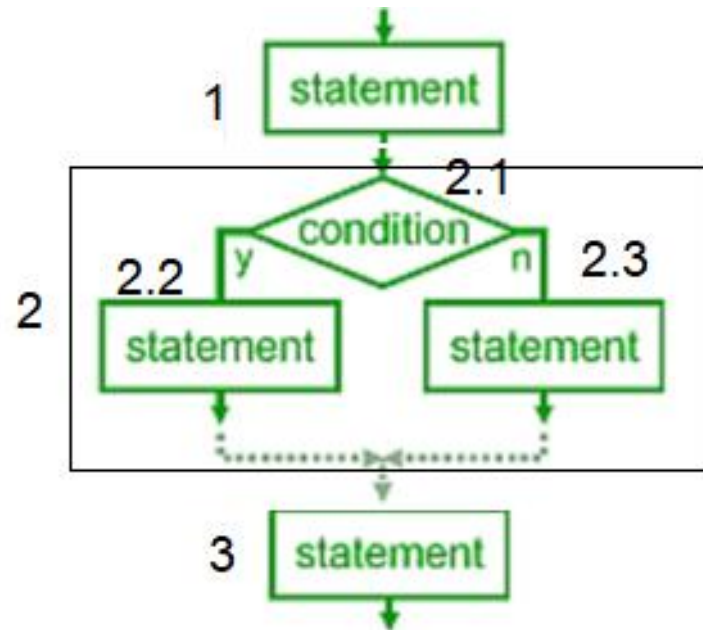
```

1
2 program PopulationCheck;
3 var
4     x: integer;
5
6 begin
7     readln ( x );
8
9     if x >= 0 then
10        begin
11            writeln('x is positive or zero');
12            writeln('Value of x = ', x);
13        end
14    else
15        begin
16            writeln('x is negative');
17            writeln('Value of x = ', x);
18        end;
19
20    writeln('after if-then-else');
21
22
23 end.
24

```

CheckX.pas

```
if condition then  
    statement1  
else  
    statement2;
```

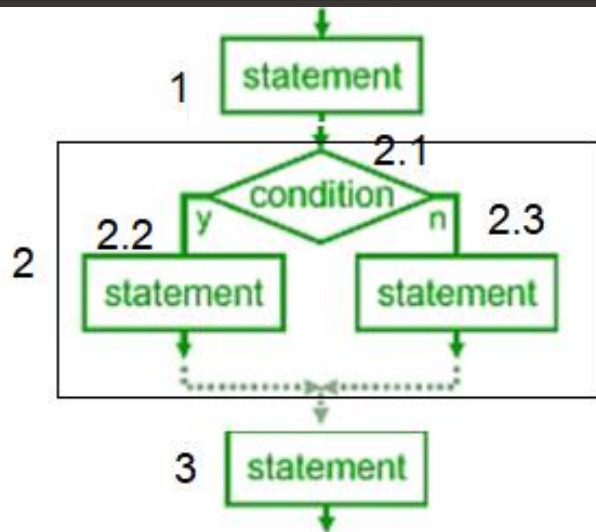


```
1  
2 program CheckX;  
3 var  
4   x: integer;  
5  
6 begin  
7   readln ( x );  
8  
9   if x >= 0 then  
10      writeln('x is positive or zero')  
11   else  
12      writeln('x is negative');  
13  
14   writeln('after if-then-else');  
15  
16  
17 end.
```

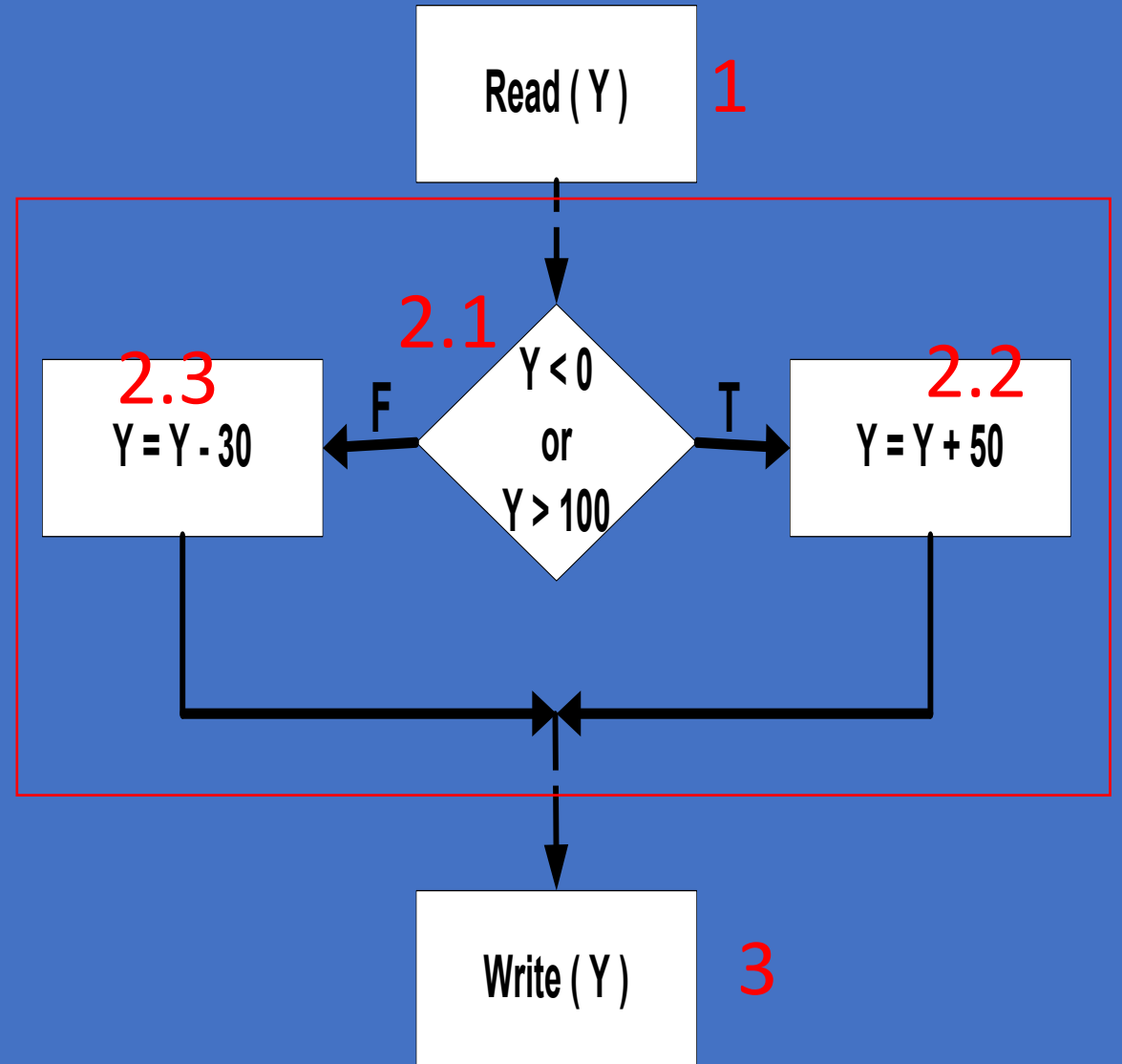
```

1  program CheckY;
2  var
3      Y: Integer;
4  begin
5      Write('Y = ');
6      ReadLn(Y);
7
8      if (Y < 0) or (Y > 100) then
9          Y := Y + 50
10     else
11         Y := Y - 30;
12
13     WriteLn(Y);
14 end.
15

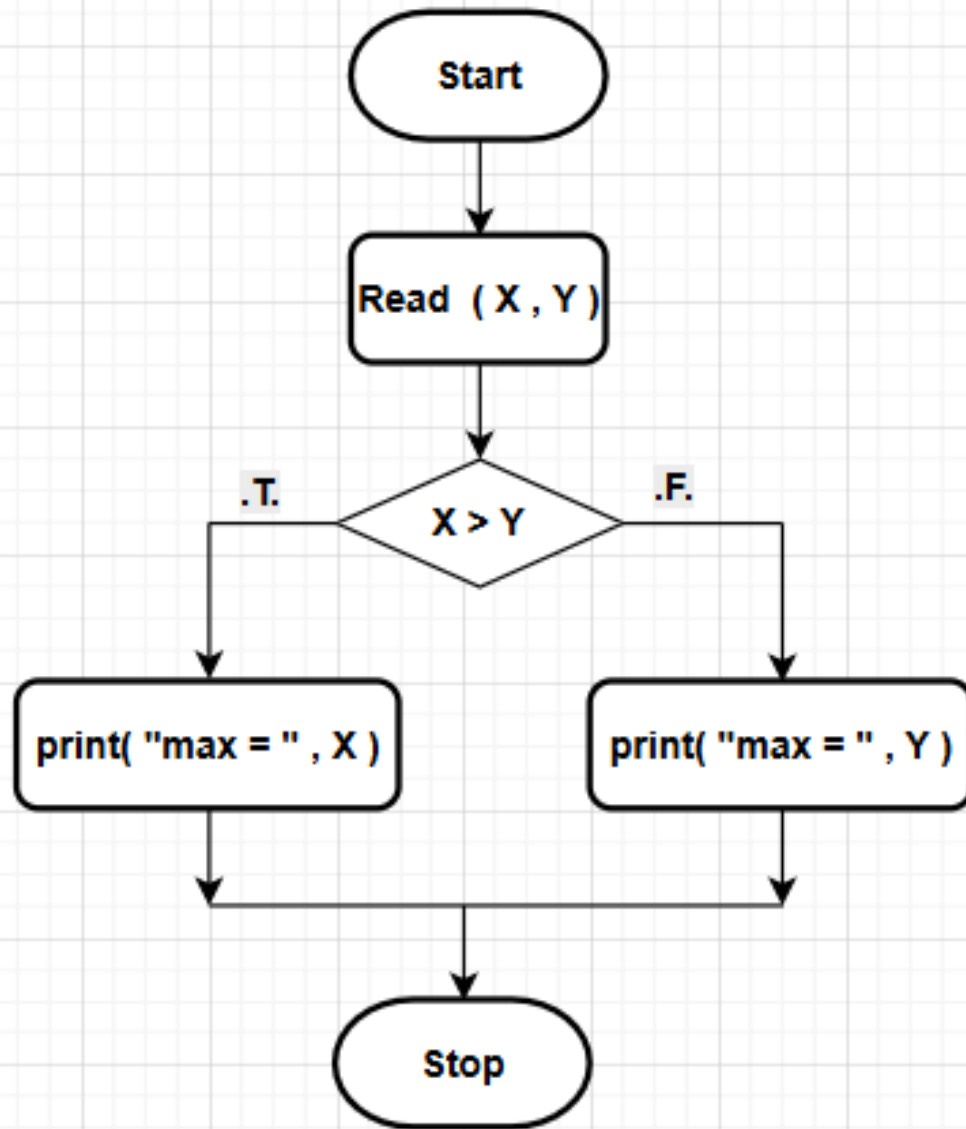
```



CheckY.pas



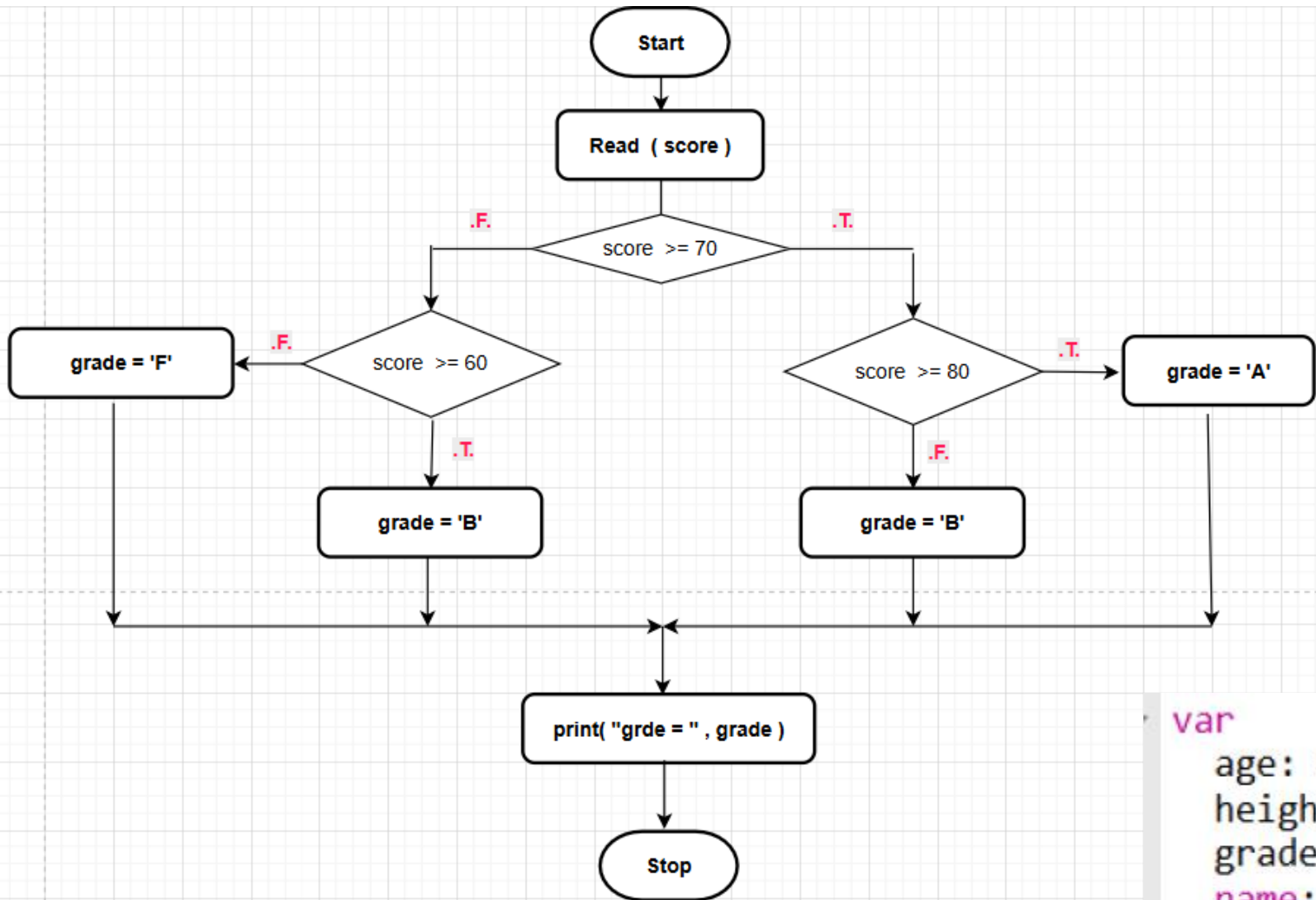
Convert to Pascal



```
program maxM3;  
var  
  
begin  
  
end.
```

computgrad.pas

Convert to Pascal



```
program maxM4;  
var  
  
begin  
  
end.
```

var

```
age: integer;  
height: real;  
grade: char;  
name: string;  
passed: boolean;
```

```
{ จำนวนเต็ม }  
{ จำนวนจริง (ทศนิยม) }  
{ ตัวอักษรหนึ่งตัว }  
{ ข้อความหรือสตริง }  
{ ค่าตรรกะ true/false }
```

Nested if-else

ตัวอย่างที่ 7.5

จงเขียนโปรแกรมรับราคาสินค้าหนึ่งรายการถ้าจำนวนราคาตั้งแต่ 1000 บาทขึ้นไป
ไปคิดภาษี 7 เปอร์เซ็นต์ หากน้อยกว่านั้นให้คิดภาษี 3 เปอร์เซ็นต์

var

age: integer;

height: real;

grade: char;

name: string;

passed: boolean;

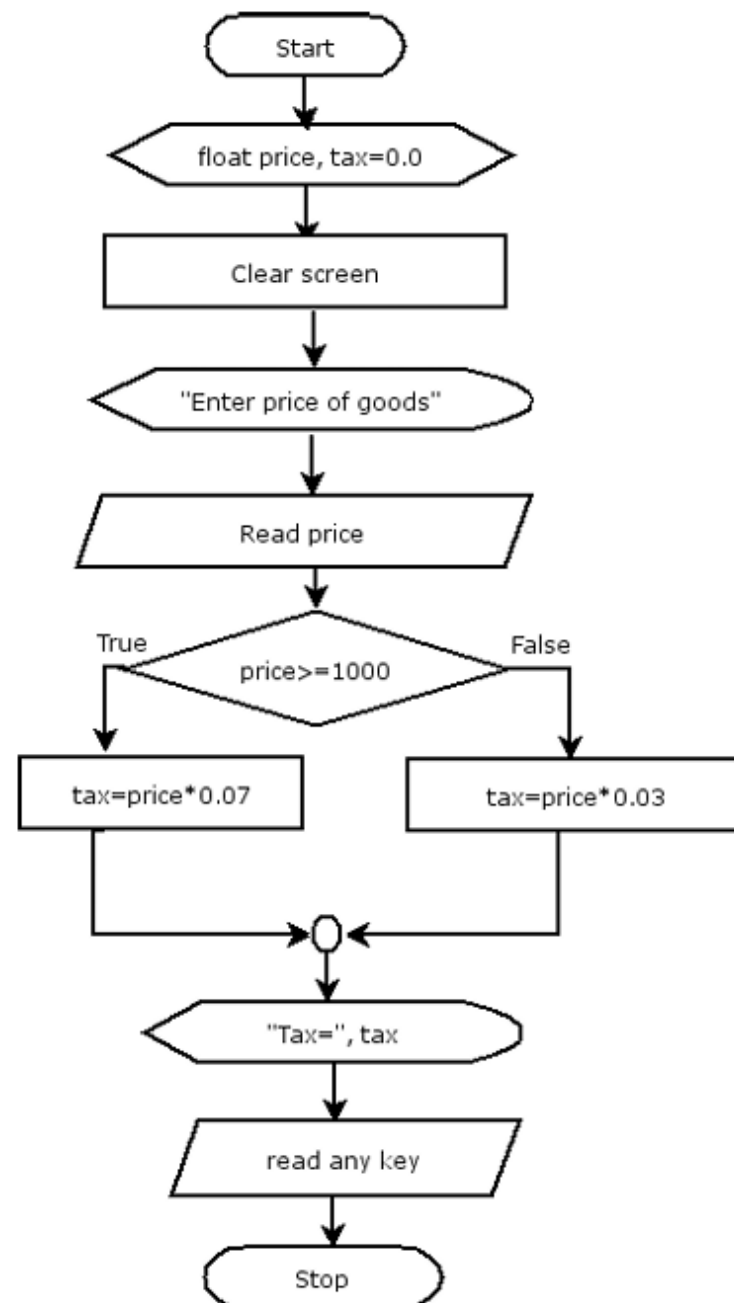
{ จำนวนเต็ม }

{ จำนวนจริง (ทศนิยม) }

{ ตัวอักษรหนึ่งตัว }

{ ข้อความหรือสตริง }

{ ค่าตรรกะ true/false }



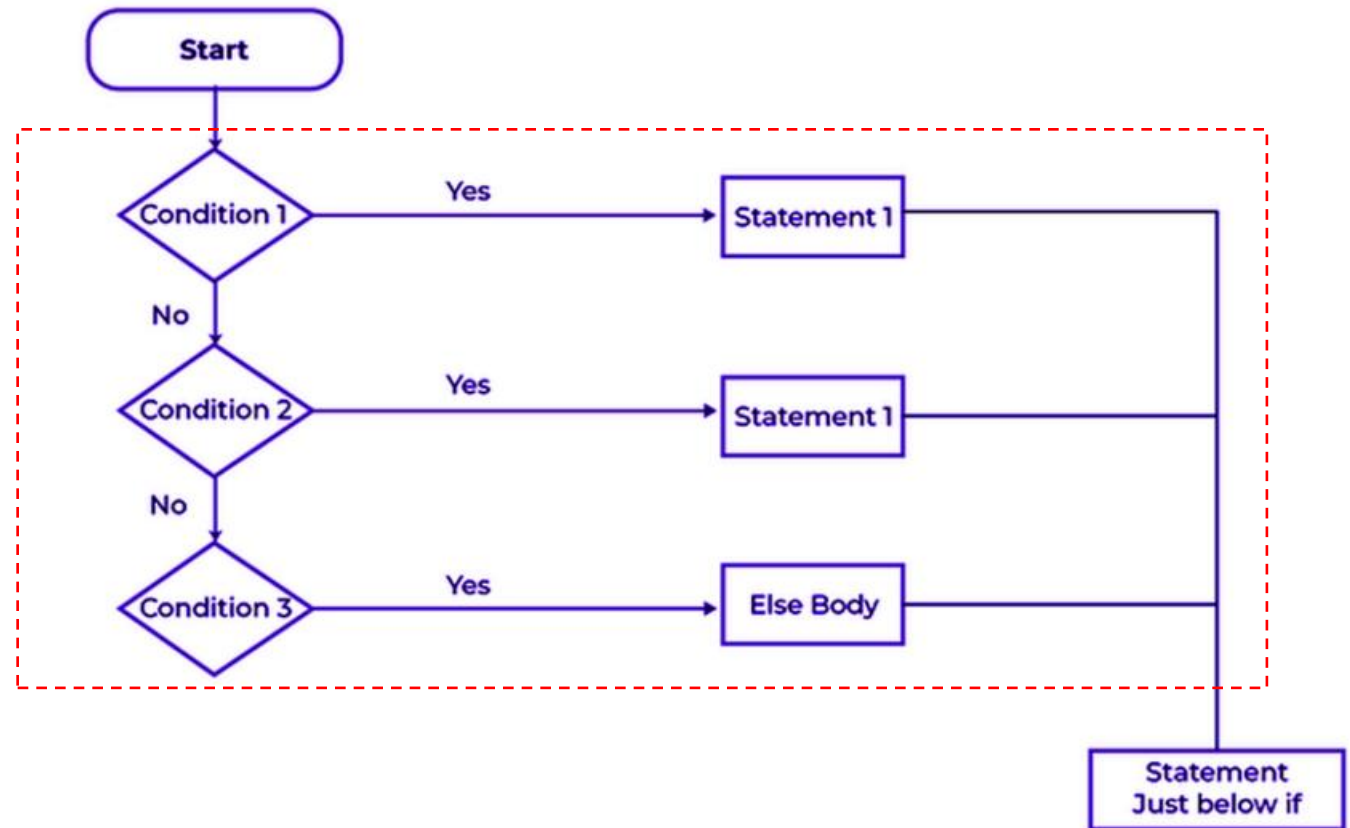
Ladder if Statement

- ❑ The if structures can be placed with other if structure., called *Ladder*

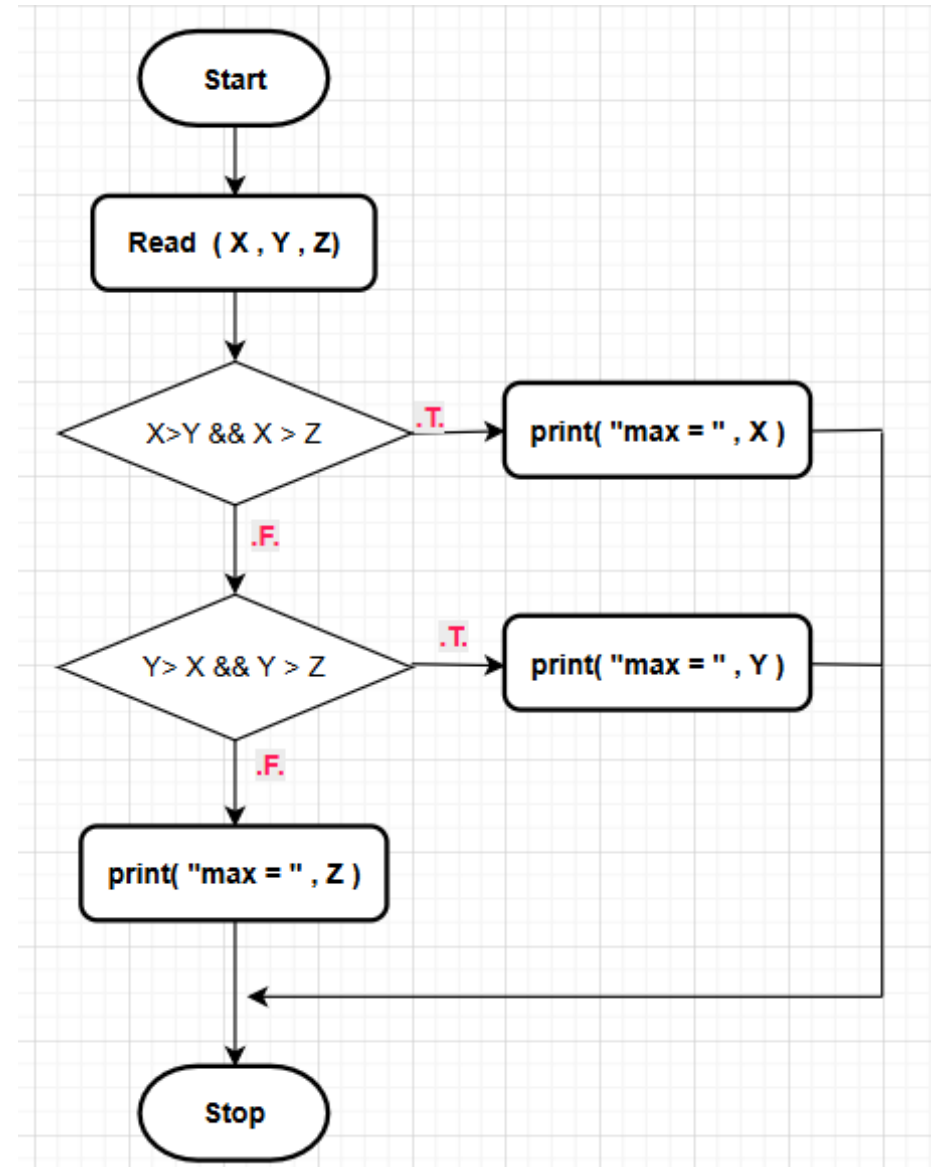
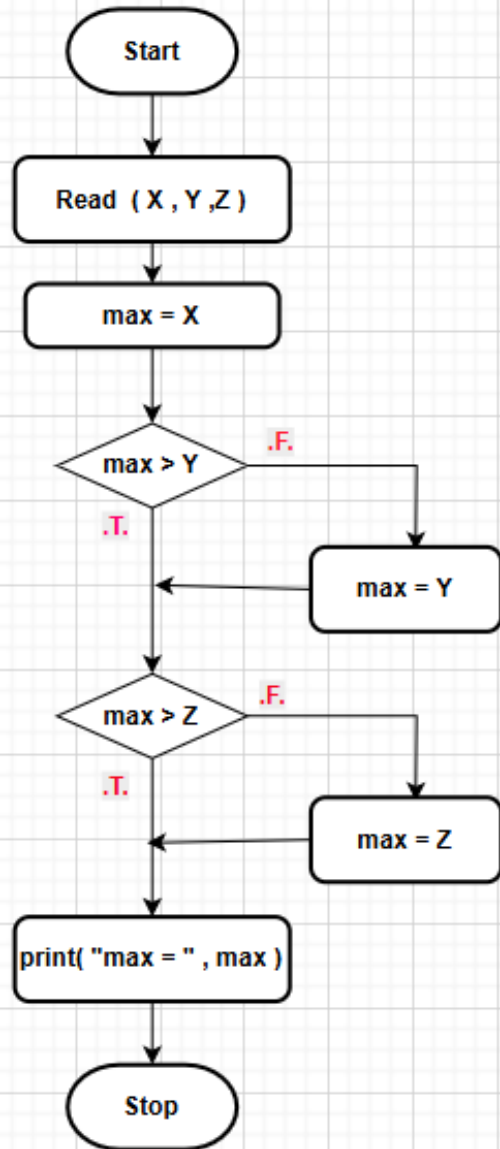
pascal

```
if condition1 then  
    statement1  
else if condition2 then  
    statement2  
else if condition3 then  
    statement3  
...  
else  
    statementN;
```

WAVE™

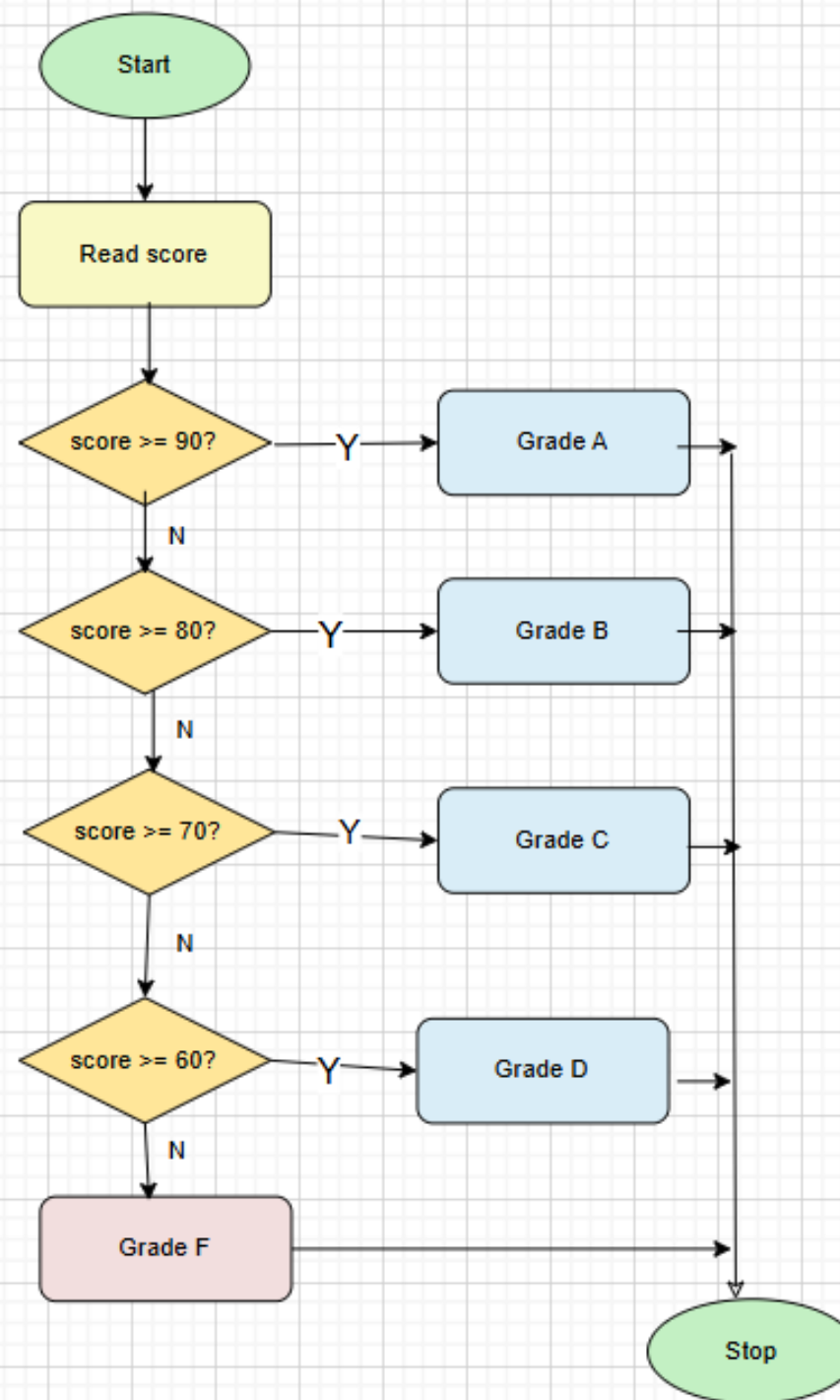


Which one is the “ladder-style decision flow” ?



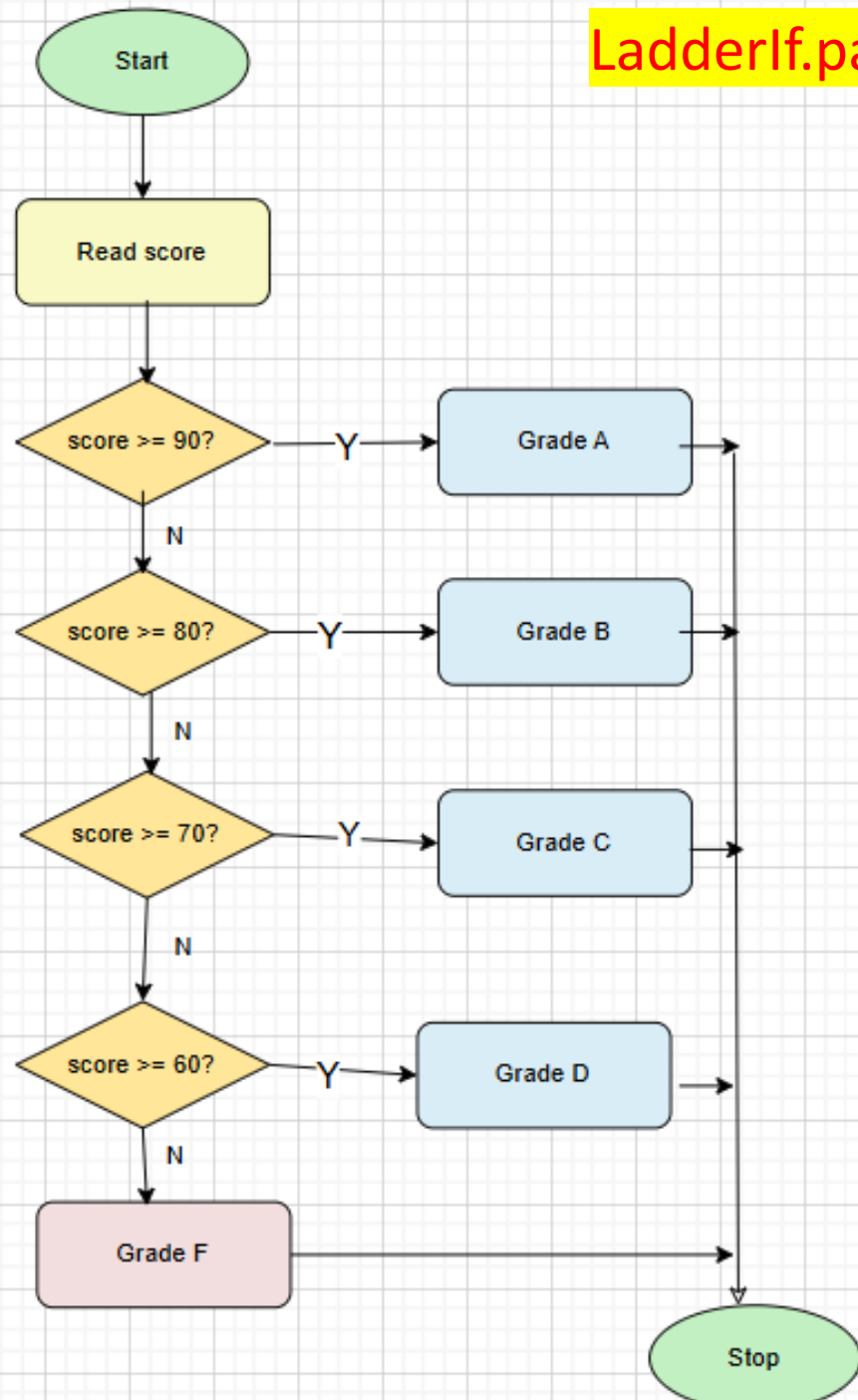
LadderIf.pas

```
if condition1 then  
    statement1  
else if condition2 then  
    statement2  
else if condition3 then  
    statement3  
...  
else  
    statementN;
```



```
1  
2 program LadderIfExample;  
3  
4 var  
5     score: integer;  
6  
7 begin  
8     Write('Enter your score: ');  
9     ReadLn(score);  
10  
11     if score >= 90 then  
12         WriteLn('Grade A')  
13     else if score >= 80 then  
14         WriteLn('Grade B')  
15     else if score >= 70 then  
16         WriteLn('Grade C')  
17     else if score >= 60 then  
18         WriteLn('Grade D')  
19     else  
20         WriteLn('Grade F');  
21  
22 end.  
23  
24
```

LadderIf.pas



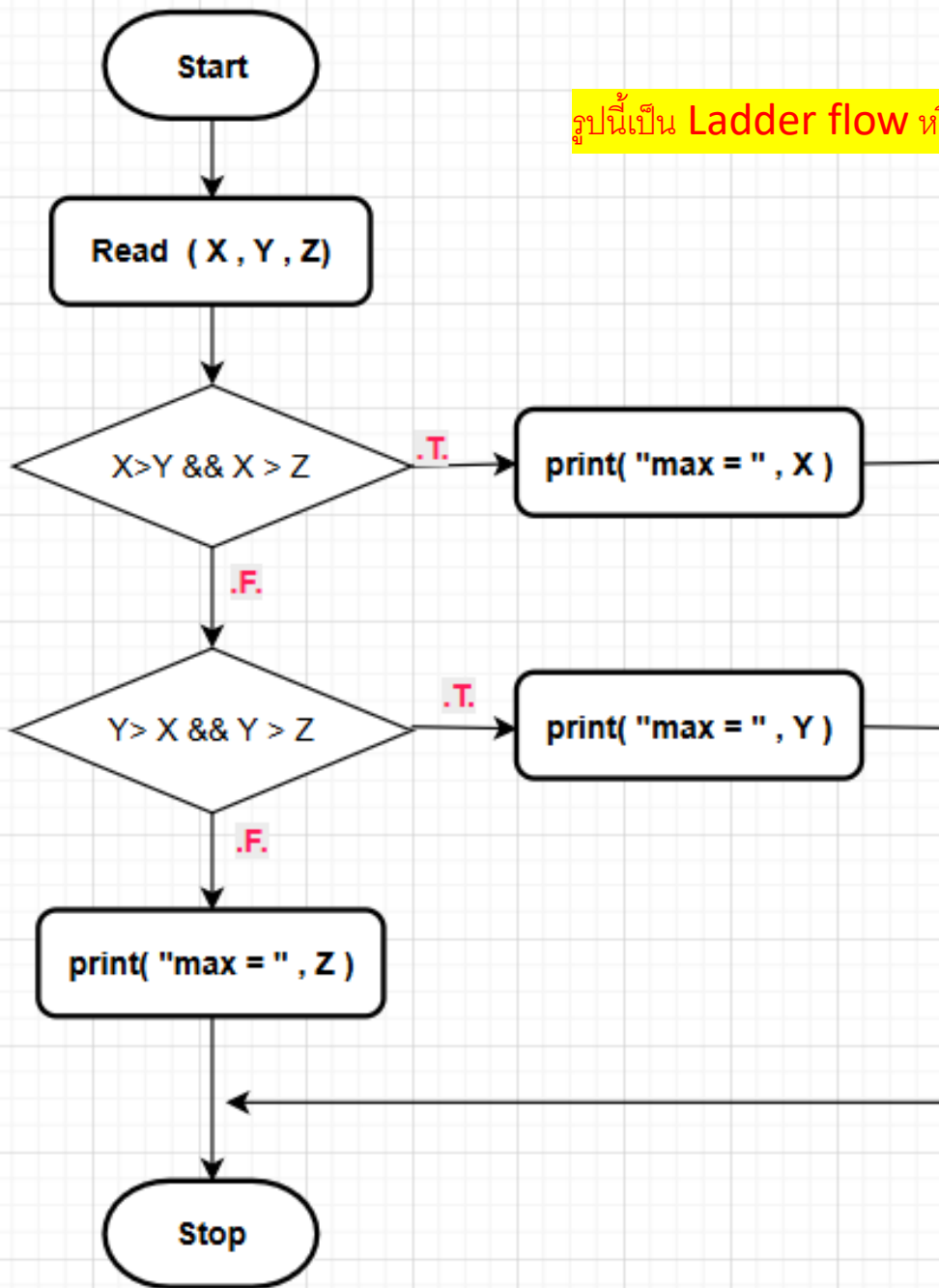
```
1
2 program LadderIfExample;
3
4 var
5     score: integer;
6
7 begin
8     Write('Enter your score: ');
9     ReadLn(score);
10
11     if score >= 90 then
12         WriteLn('Grade A')
13     else if score >= 80 then
14         WriteLn('Grade B')
15     else if score >= 70 then
16         WriteLn('Grade C')
17     else if score >= 60 then
18         WriteLn('Grade D')
19     else
20         WriteLn('Grade F');
21
22 end.
23
24
```

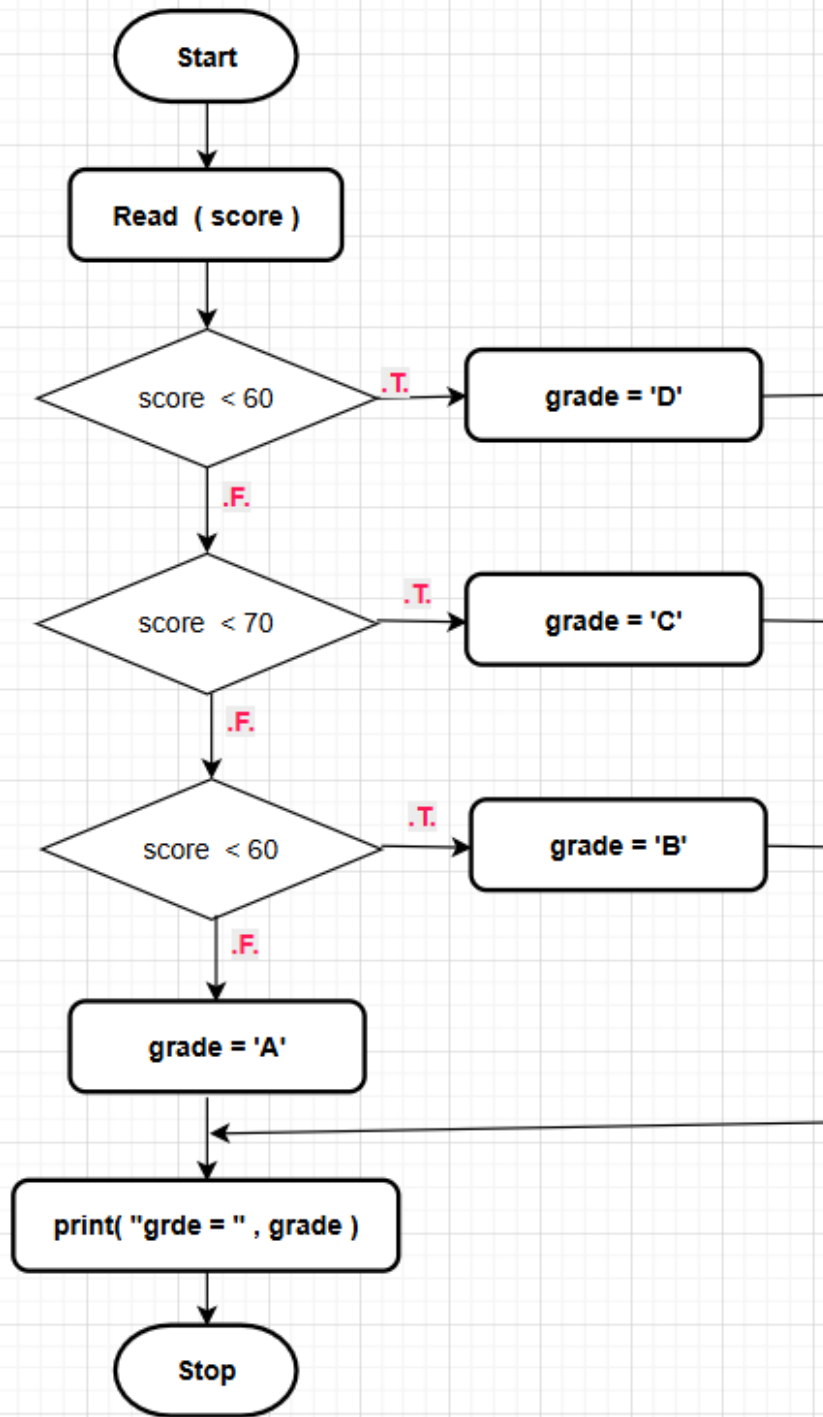
รูปนี้เป็น Ladder flow หรือไม่

Convert to Pascal

```
program maxM5;  
var  
  
begin  
  
end.
```

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
and	และ (and,&&)	$x > 5$ and $y < 10$	false
or	หรือ (or,)	$x == 0$ or $y == 0$	false
not	ไม่ (not,!)	not($x == 5$)	True





Convert to Pascal



HOMEWORK 2

จงเขียน โปรแกรมรับเวลาเข้า/ออก และคำนวณค่าจอดรถยนต์ในห้างสรรพสินค้า

ข้อกำหนด

1. ห้างสรรพสินค้าเปิดให้บริการตั้งแต่เวลา 9.30-22.00 ไม่อนุญาตให้จอดค้างคืน

2. เงื่อนไขการคิดเงิน

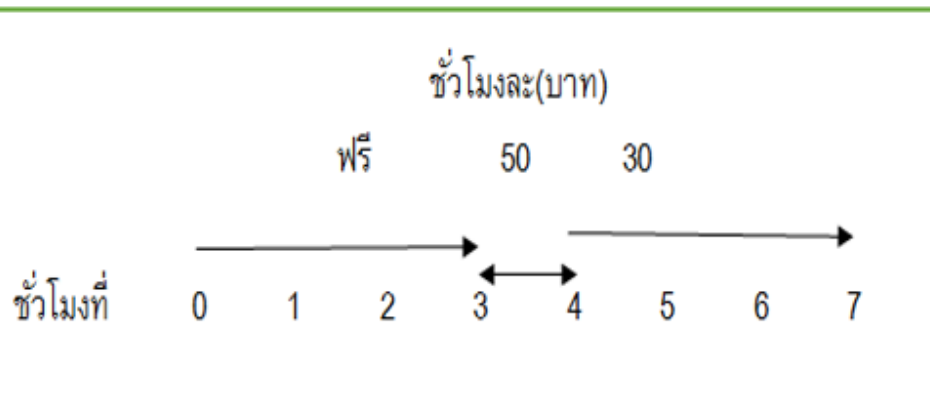
2.1. อัตราค่าจอดรถยนต์ 2 ชั่วโมงแรกฟรี

2.2. ชั่วโมงที่ 4 คิดชั่วโมงละ 50 บาท

2.3. ชั่วโมง ต่อๆ ไปคิดค่าจอดชั่วโมงละ 30 บาท

โดยเศษของชั่วโมงคิดเป็น 1 ชั่วโมง

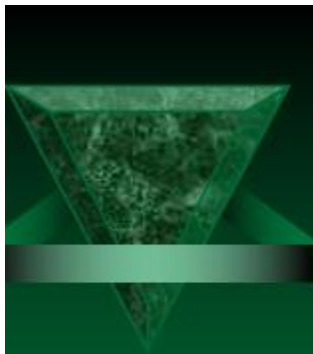
ดังรูป



วิเคราะห์การประมวลผล

1. รับเวลาออก – เวลาเข้าจอด
2. คำนวณจำนวนนาฬิกาที่จอด (M_total)
3. แปลงจำนวนนาฬิกา (M_total) ให้เป็นจำนวน ชั่วโมง (H_total) ถ้ามีเศษนาฬิกา ให้นับชั่วโมงเพิ่มขึ้น 1
4. นำชั่วโมงที่ถูกคำนวณมาเปรียบเทียบกับเงื่อนไขตามข้อกำหนด

```
put Time-in :  
  hour  =? 13  
  minute=? 30  
put Time-out :  
  hour  =? 18  
  minute=? 23  
-----  
Time in : 13:30  
Time out: 18:23  
Parking time    = 5 hours  
Service charge  = 80 Baths
```

DATA REQUIREMENTS

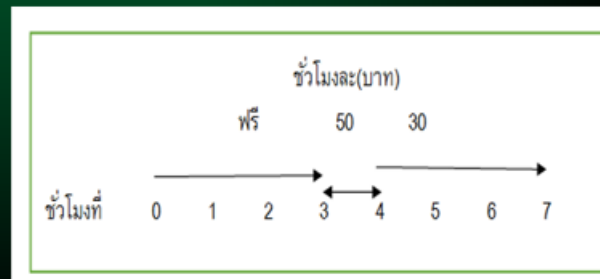
H_in ชั่วโมงเข้าจอด

M_in นาทีเข้าจอด

H_out ชั่วโมงออก

M_out นาทีออก

Fee ค่าธรรมเนียม



การคำนวณนาที

1. 4.40 - 4.49 (9 นาที)

2. 4.40 - 5.12 (20+12 = 32 นาที)

3. 4.40 - 5.40 (20+40 = 60 นาที)

4. 4.40 - 5.45 (20+45 = 65 นาที)

6. 4.40 - 6.18 (20+1*60+18 = 98 นาที)

7. 4.40 - 7.45 (20+3*60+45 = 245 นาที)



1. 4.40 - 4.49 (9 นาที โดยที่ $(4-4-1)*60 < 0$)

2. 4.40 - 5.12 (20+(5-4-1)*60+12 = 32 นาที)

3. 4.40 - 5.40 (20+(5-4-1)*60+40 = 60 นาที)

4. 4.40 - 5.45 (20+(5-4-1)*60+45 = 65 นาที)

6. 4.40 - 6.18 (20+(6-4-1)*60+18 = 98 นาที)

7. 4.40 - 7.45 (20+ (7-4-1)*60 +45 = 245 นาที)

H_out

H_in

M_out

7. 4.40 - 7.45 (20+ (7-4-1)*60 +45 = 245 นาที)

การคำนวณนาที

1. 4.40 - 4.49 (9 นาที)

2. 4.40 - 5.12 (10+12 = 32 นาที)

3. 4.40 - 5.40 (20+40 = 60 นาที)

4. 4.40 - 5.45 (20+45 = 65 นาที)

6. 4.40 - 6.18 (20+1*60+18 = 98 นาที)

7. 4.40 - 7.45 (20+2*60+45 = 185 นาที)



1. 4.40 - 4.49 (9 นาที โดยที่ $(4-4-1)*60 < 0$)

2. 4.40 - 5.12 (20+(5-4-1)*60+12 = 32 นาที)

3. 4.40 - 5.40 (20+(5-4-1)*60+40 = 60 นาที)

4. 4.40 - 5.45 (20+(5-4-1)*60+45 = 65 นาที)

6. 4.40 - 6.18 (20+(6-4-1)*60+18 = 98 นาที)

7. 4.40 - 7.45 (20+ (7-4-1)*60 +45 = 185 นาที)

1. 4.40 - 4.49 (9 นาที โดยที่ $(4-4) = 0$)

2. 4.40 - 5.12 ($(60-40)+(5-4-1)*60+12 = 32$ นาที)

3. 4.40 - 5.40 ($(60-40) +(5-4-1)*60+40 = 60$ นาที)

4. 4.40 - 5.45 ($(60-40) +(5-4-1)*60+45 = 65$ นาที)

6. 4.40 - 6.18 ($(60-40) +(6-4-1)*60+18 = 98$ นาที)

7. 4.40 - 7.45 ($((60-40) + (7-4-1)*60 +45 = 185$ นาที)

H_in

ชั่วโมงเข้าจอด

M_in

นาทีเข้าจอด

H_out

ชั่วโมงออก

M_out

นาทีออก

M_in

H_out

H_in

M_out

7. 4.40 - 7.45 (60-40)+ (7-4-1)*60 +45 = 185 นาที)

general case :

7. 4.40 - 7.45

$$\begin{array}{ccccccc} \text{M_in} & & \text{H_out} & & \text{H_in} & & \text{M_out} \\ & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \\ & (60-40) & + & (7-4-1)*60 & + & 45 & = 185 \text{ นาที} \\ & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & & \\ & \text{Diff_Min} & & \text{Diff_H} & & & \text{M_total} \\ & 20 & & 120 & & & 45 \end{array}$$

special case :

1. 4.40 - 4.49 (9 นาที โดยที่ $(4-4) = 0$)

If $(\text{H_out} - \text{H_in} == 0)$

$\text{M_total} = \text{M_out} - \text{M_in}$

Else

$\text{M_total} = \text{Diff_min} + \text{Diff_H} + \text{M_out}$

Where :

$\text{Diff_H} = (\text{H_out} - \text{H_in} - 1) * 60$

$\text{Diff_Min} = 60 - \text{M_in}$



แปลงจำนวนนาที (M_total) ให้เป็นจำนวนชั่วโมง (H_total)

general case :

$$7.40 - 7.45 \quad \begin{array}{c} \text{M}_{in} \\ \swarrow \\ (60-40) + \end{array} \begin{array}{c} \text{H}_{out} \\ \swarrow \\ (7-4-1)*60 + \end{array} \begin{array}{c} \text{H}_{in} \\ \swarrow \\ 45 = \end{array} \begin{array}{c} \text{M}_{out} \\ \swarrow \\ 245 \text{ นาที} \end{array}$$

$\underbrace{(60-40)}_{\text{Diff_Min}} \quad \underbrace{(7-4-1)*60}_{\text{Diff_H}} \quad \underbrace{45}_{\text{M_total}}$

$$H_total = H_total_1 + H_total_2$$

where

1. $H_total_1 = M_total/60$

2. if $M_total \% 60 == 0$

$H_total_2 = 0$

else

$H_total_2 = 1$

main.pas

```
1 program ParkingFeeCalculator;
2 var
3     H_in, M_in: integer;
4     H_out, M_out: integer;
5     Diff_Min, Diff_H: integer;
6     M_total, H_total: integer;
7     M_fraction, H_total_1, H_total_2: integer;
8     fee: integer;
9 begin
10    WriteLn('Put Time-in :');
11    Write('  hour    =? '); ReadLn(H_in);
12    Write('  minute =? '); ReadLn(M_in);
13
14    WriteLn('Put Time-out :');
15    Write('  hour    =? '); ReadLn(H_out);
16    Write('  minute =? '); ReadLn(M_out);
17
18    Diff_H := (H_out - H_in - 1) * 60;
19
20    if (Diff_H < 0) then
21        M_total := M_out - M_in
22    else
23        begin
24            Diff_Min := 60 - M_in;
25            M_total := Diff_Min + Diff_H + M_out;
26        end;
```

ParkingFee.pas

```
28     H_total_1 := M_total div 60;
29     H_total_2 := 0;
30     M_fraction := M_total mod 60;
31
32     if (M_fraction <> 0) then
33     begin
34         H_total_2 := 1;
35     end;
36
37     H_total := H_total_1 + H_total_2;
38
39     WriteLn;
40     WriteLn('Time in   : ', H_in, ':', M_in);
41     WriteLn('Time out  : ', H_out, ':', M_out);
42     WriteLn('Total minute : ', M_total);
43     WriteLn('Total hour   : ', H_total_1);
44     WriteLn('Fraction minute : ', M_fraction);
45     WriteLn('Park hour    : ', H_total);
46
47     if (H_total <= 3) then
48         fee := 0
49     else if (H_total <= 4) then
50         fee := 50 * (H_total - 3)
51     else
52         fee := 50 * 1 + 30 * (H_total - 4);
53
54     WriteLn('Fee : ', fee);
55 end.
```

```
56 lines compiled, 0.0 sec
```

```
Put Time-in :
```

```
hour   =? 4
```

```
minute =? 21
```

```
Put Time-out :
```

```
hour   =? 7
```

```
minute =? 32
```

```
Time in   : 4:21
```

```
Time out  : 7:32
```

```
Total minute : 191
```

```
Total hour   : 3
```

```
Fraction minute : 11
```

```
Park hour    : 4
```

```
Fee : 50
```

```
56 lines compiled, 0.0 sec
```

```
Put Time-in :
```

```
hour   =? 17
```

```
minute =? 21
```

```
Put Time-out :
```

```
hour   =? 18
```

```
minute =? 19
```

```
Time in   : 17:21
```

```
Time out  : 18:19
```

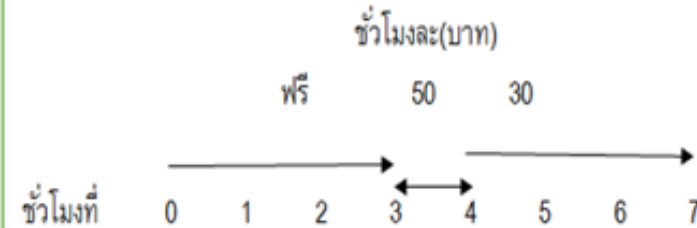
```
Total minute : 58
```

```
Total hour   : 0
```

```
Fraction minute : 58
```

```
Park hour    : 1
```

```
Fee : 0
```



```
56 lines compiled, 0.0 sec
```

```
Put Time-in :
```

```
hour   =? 11
```

```
minute =? 11
```

```
Put Time-out :
```

```
hour   =? 18
```

```
minute =? 9
```

```
Time in   : 11:11
```

```
Time out  : 18:9
```

```
Total minute : 418
```

```
Total hour   : 6
```

```
Fraction minute : 58
```

```
Park hour    : 7
```

```
Fee : 140
```

HOME WORK 1

“ จงเขียนโปรแกรมรับ ยอดขายของพนักงานขาย และฐานเงินเดือน จาก **keyboard** เพื่อคำนวณ เงินเดือนของพนักงานขาย โดยมีสูตรดังนี้

$$\text{เงินเดือน} = \text{ฐานเงินเดือน} + \text{โบนัส}$$

เงื่อนไขการคำนวณโบนัส

- ถ้ายอดขายเกิน 10,000 บาท เพิ่มเงินโบนัสให้ 10 % ของยอดขาย
- ถ้ายอดต่ำกว่าหรือ เท่ากับ 10,000 บาท ให้โบนัส 5 % ของยอดขาย

HOME WORK 2

จงเขียนโปรแกรมคิดค่าน้ำ โดยป้อนเลขที่มิเตอร์ใหม่ และมิเตอร์
เดือนที่แล้วทางแป้นพิมพ์ หาค่าน้ำที่ต้องจ่ายจากอัตราดังนี้

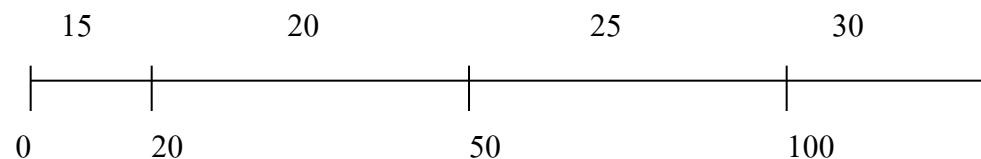
จำนวนมิเตอร์ที่ใช้(ยูนิต)	อัตราจ่าย(บาทต่อยูนิต)
---------------------------	------------------------

0-20	15
------	----

21-50	ส่วนที่เกิน 20 คิดยูนิตละ 20 บาท
-------	----------------------------------

51-100	ส่วนที่เกิน 50 คิดยูนิตละ 25 บาท
--------	----------------------------------

>100	ส่วนที่เกิน 100 คิดยูนิตละ 30 บาท
------	-----------------------------------



แบบฝึกหัด 4

จงเขียนโปรแกรมรับวัน เดือน ปี ปัจจุบัน และวัน เดือน ปี วันเกิดของบุคคลใดๆ

จงหาอายุของบุคคลคนนี้ว่ามีอายุเท่ากับกี่ปี กี่เดือน กี่วัน

กำหนดให้ 1 ปี เท่ากับ 12 เดือน

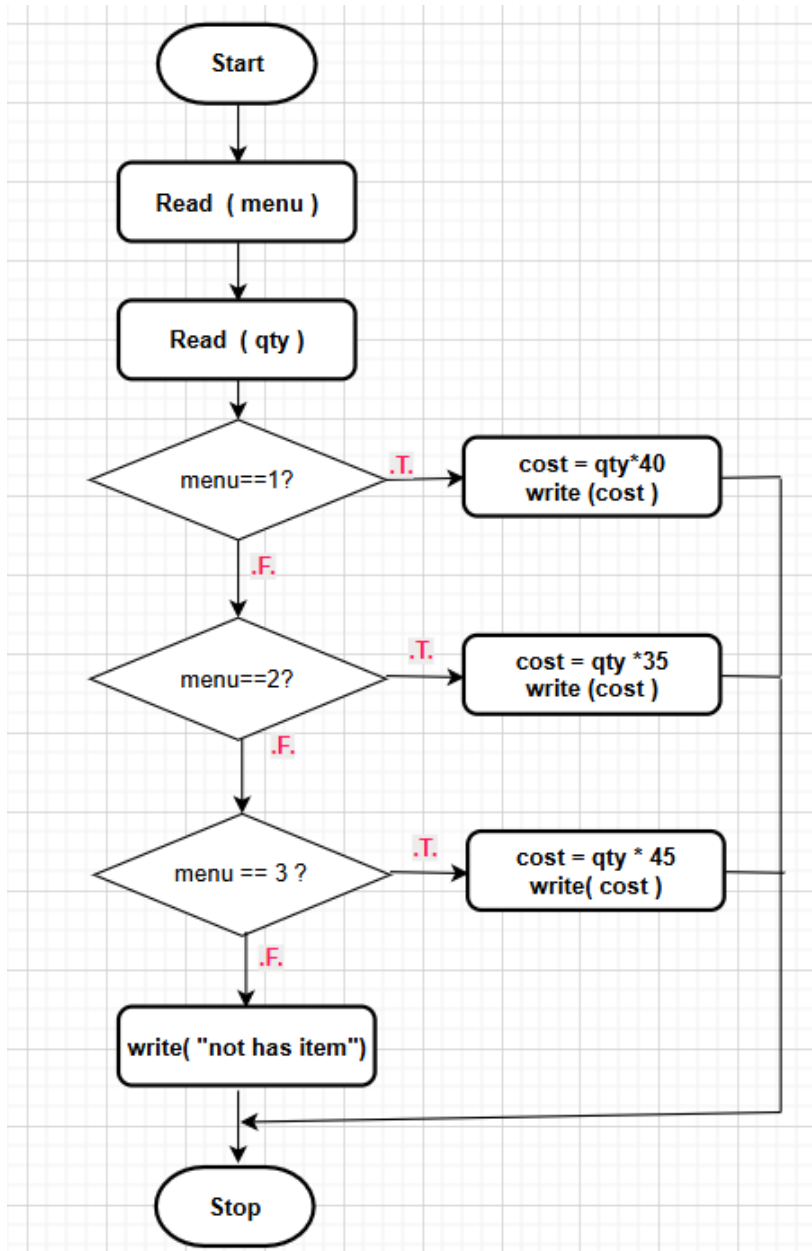
1 เดือน เท่ากับ 30 วัน

5.2. Selection Structure

case of statement

(multi-way structure)

จงเขียนโปรแกรมให้ผู้ใช้เลือกเมนูอาหาร (1-3) และใส่จำนวนจานที่ต้องการ จากนั้นคำนวณ ราคารวม



```
1  program SwitchExample;
2  var
3      menu: integer;
4      qty : integer;
5      cost : real;
6  begin
7      write( 'menu =');
8      readln(menu);
9      write( 'qty =');
10     readln(qty);
11     if menu = 1 then
12     begin
13         cost := qty*40.0;
14         write('cost = ', cost:0:2 );
15     end
16     else if menu = 2 then
17     begin
18         cost := qty*35.0;
19         write('cost = ', cost:0:2 );
20     end
21     else if menu = 3 then
22     begin
23         cost := qty*45.0;
24         write('cost = ', cost:0:2 );
25     end
26     else
27         writeln('not has item');
28
29 end.
```

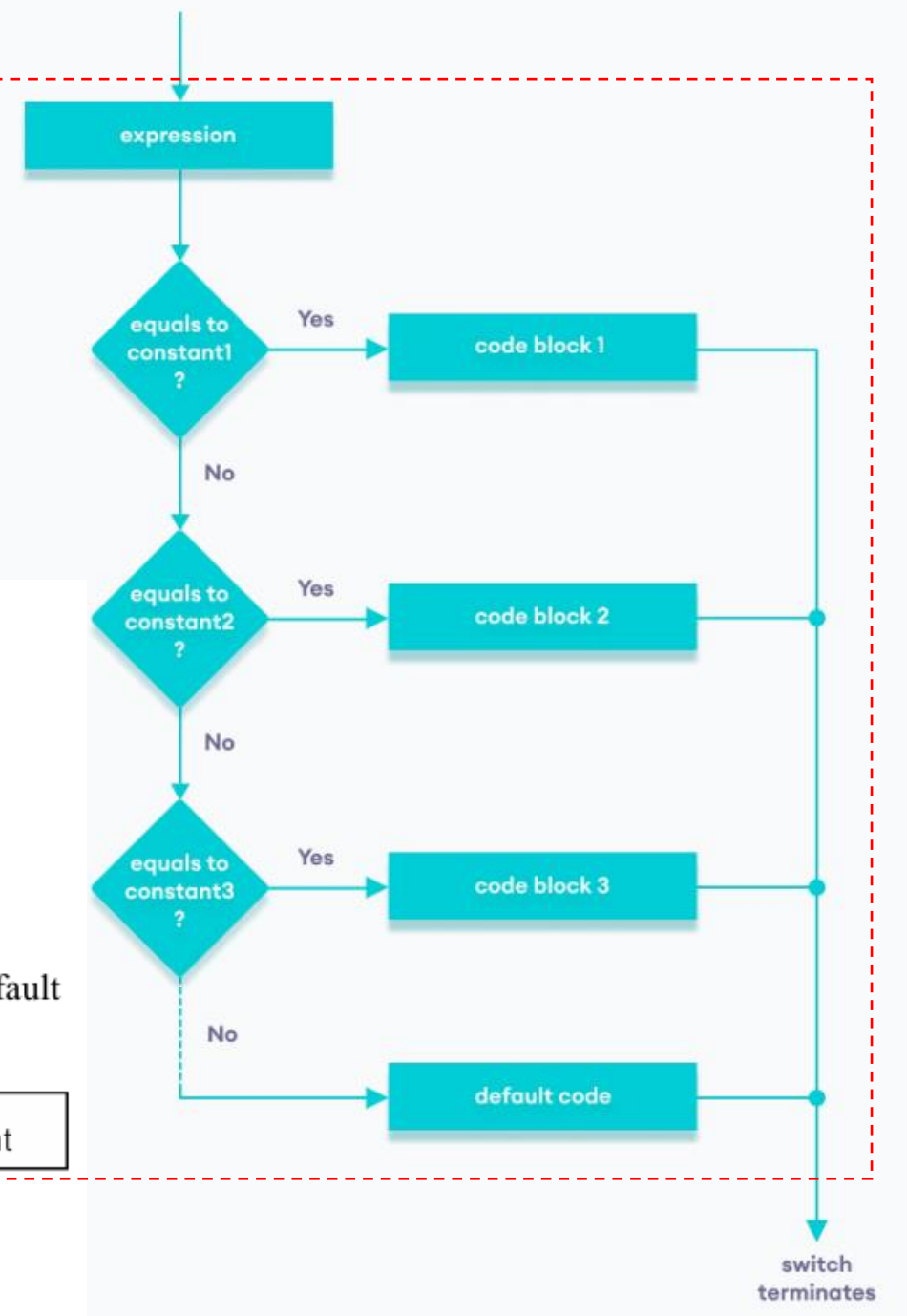
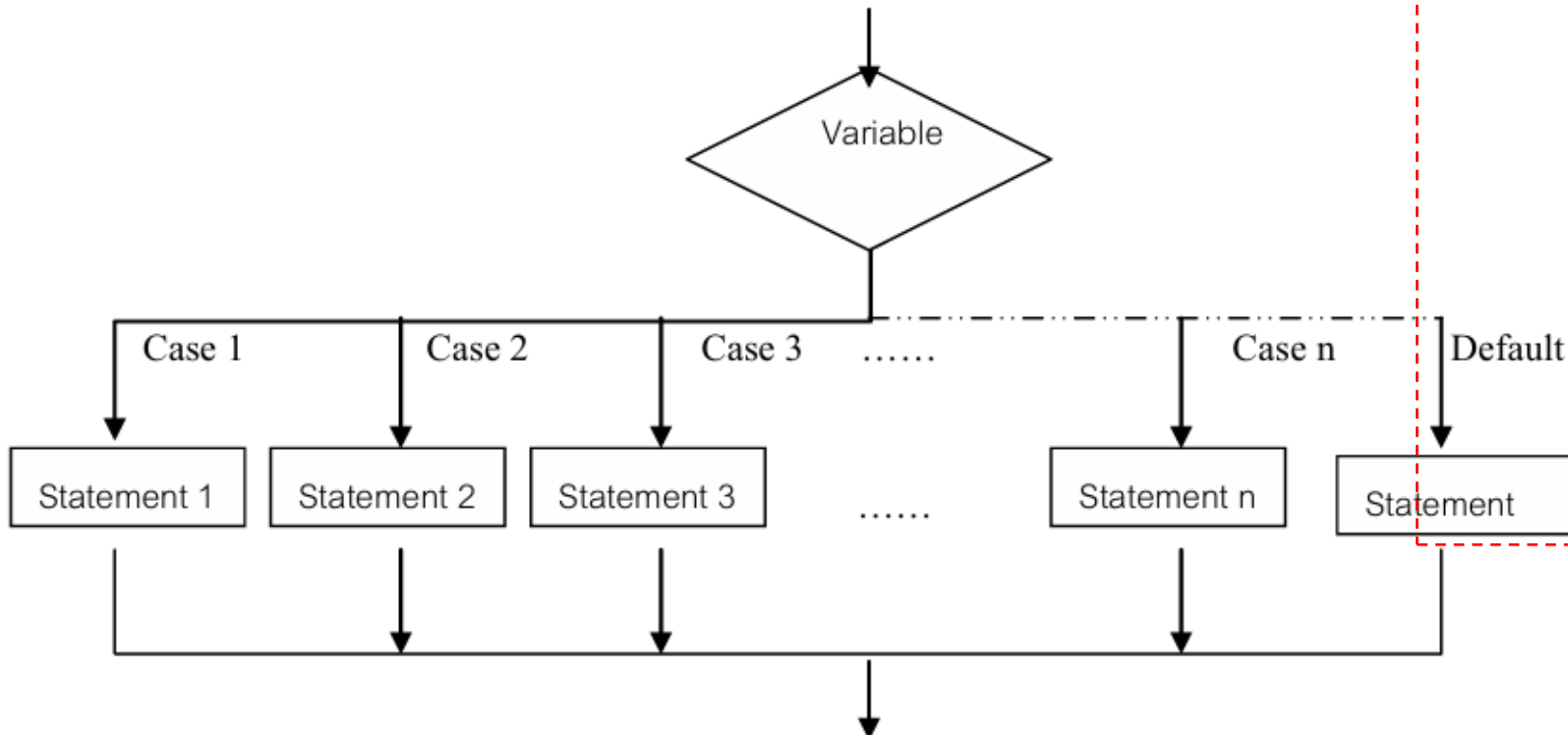
SwitchEx.pas

```

case <expression> of
  value1: statement1;
  value2: statement2;
  value3, value4: statement3; { multiple values for same statement }
else
  statementElse; { optional }
end;

```

หลักการทำงาน

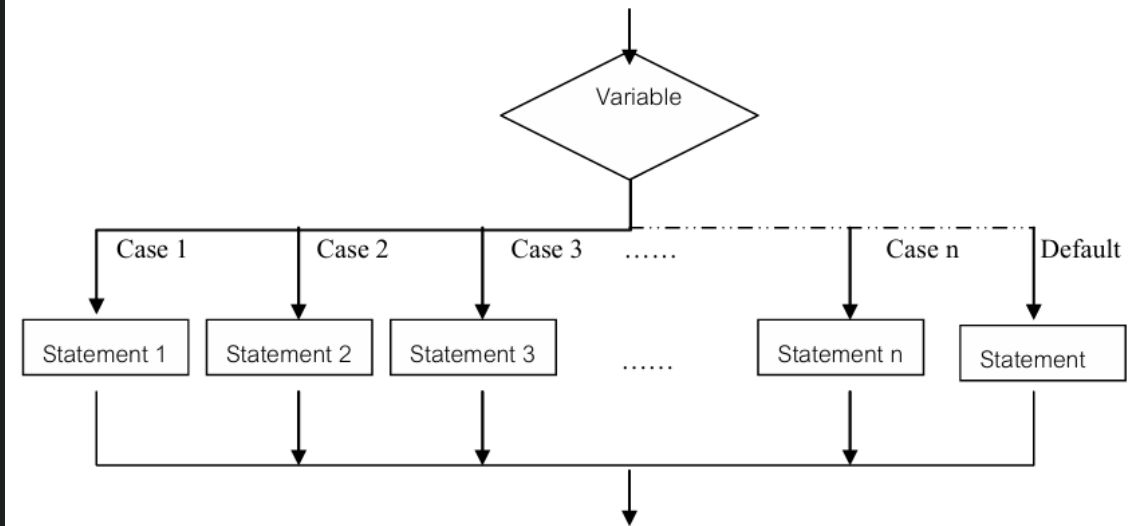


```

1  program CaseExample;
2
3  var                                Case.pas
4      grade: char;
5
6  begin
7      write('Enter grade (A/B/C/D/F): ');
8      readln(grade);
9
10     case grade of
11         'A': writeln('Excellent!');
12         'B': writeln('Good!');
13         'C': writeln('Average.');
14         'D': writeln('Below average.');
15         'F': writeln('Fail.');
16     else
17         writeln('Invalid grade.');
18     end;
19 end.
20

```

หลักการทำงาน



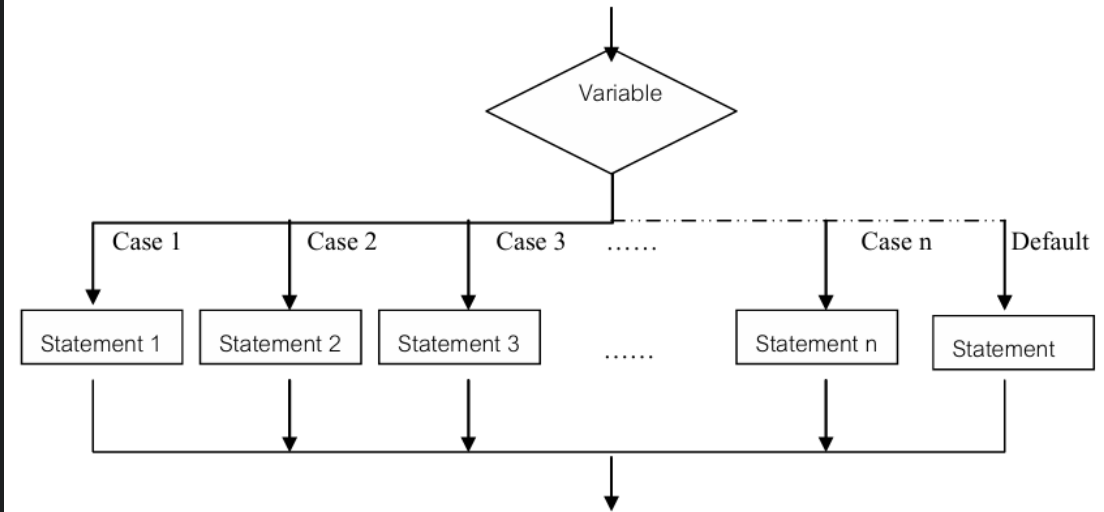
```

1  program CaseWithBlock;
2
3  var
4      day: integer;
5
6  begin
7      write('Enter day number (1-7): ');
8      readln(day);
9
10     case day of
11         1: begin
12             writeln('Monday');
13             writeln('Start of the week');
14         end;
15         5: begin
16             writeln('Friday');
17             writeln('Almost weekend!');
18         end;
19         6, 7: writeln('Weekend!');
20     else
21         writeln('Invalid day');
22     end;
23 end.
24

```

CaseWithBlock.pas

หลักการทำงาน



```

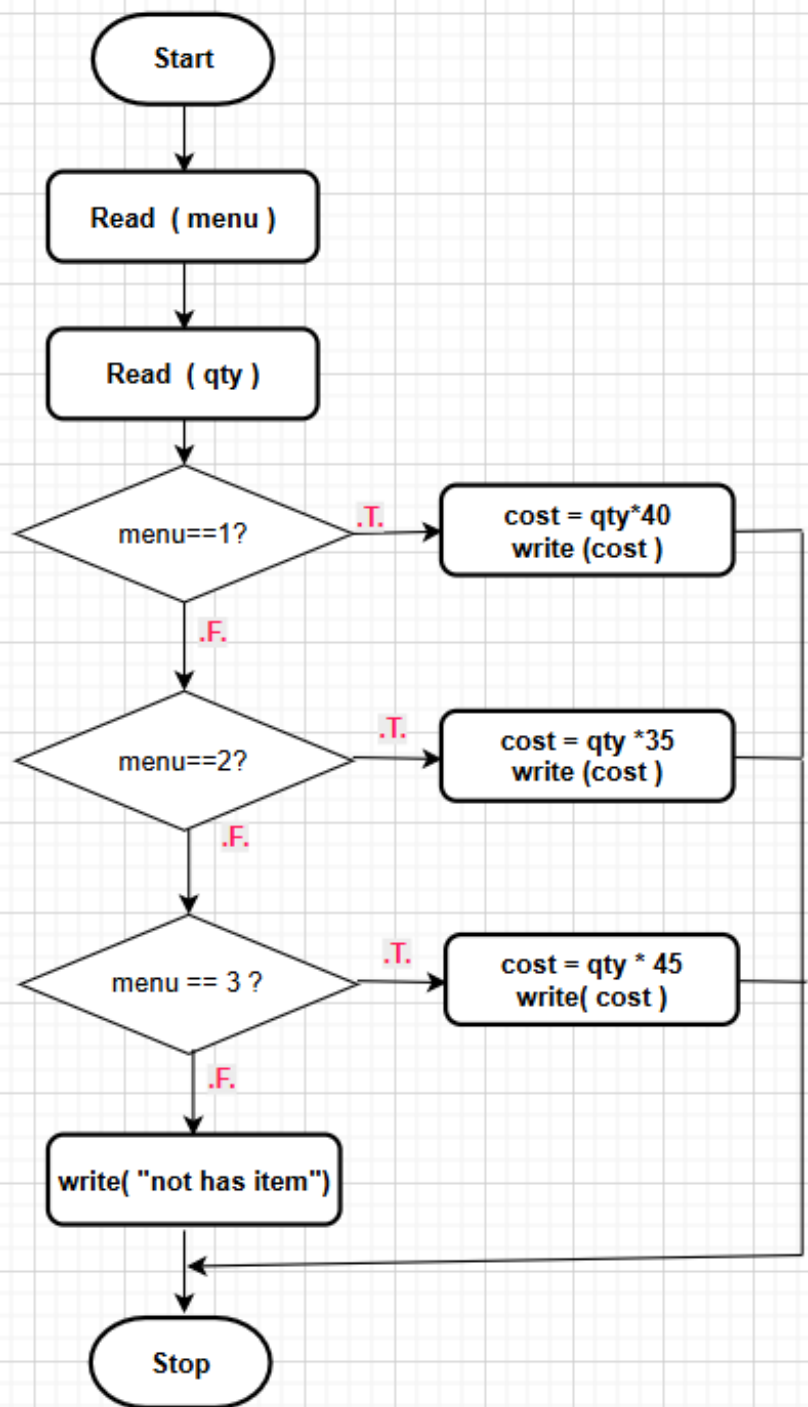
1  program SwitchExample;
2  var
3      a: integer;           SwitchExample.pas
4  begin
5      readln(a);
6
7  case a of
8      10: writeln('Value of a is 10');
9      20: writeln('Value of a is 20');
10     30: writeln('Value of a is 30');
11 else
12     writeln('Value of a is not matching');
13 end;
14
15 writeln('Exact value of a is : ', a);
16 end.

```

```

1  program IfElseExample;
2  var
3      a: Integer;           IfElseEx.pas
4  begin
5      Write('Enter a value: ');
6      ReadLn(a);
7      if a = 10 then
8          WriteLn('Value of a is 10')
9      else if a = 20 then
10         WriteLn('Value of a is 20')
11     else if a = 30 then
12         WriteLn('Value of a is 30')
13     else
14         WriteLn('Value of a is not matching');
15
16     WriteLn('Exact value of a is : ', a);
17 end.

```



Convert to PASCAL

MenuCase.pas

ตัวอย่างที่ 2.5

จงเขียนผังงานโปรแกรมของโปรแกรมให้ผู้ใช้เลือกเมนูดังนี้

เมนู 1 รับชื่อแล้วแสดงผลทางจอภาพ

เมนู 2 รับนามสกุลแล้วแสดงผลทางจอภาพ

เมนู 3 รับเงินเดือนแล้วแสดงผล

เมนู 4 จบงาน

