

LAST LECTURE - STATEMENT

คำสั่ง (Statements) คือ คำสั่งที่เขียนในโปรแกรมแต่ละบรรทัด แต่ละคำสั่งปิดด้วยเครื่องหมาย **semicolon (;)**

E.g.

```
printf("Hello!\n");
```

กลุ่มคำสั่ง (a compound statement) หรือ บล็อก (**block**) คือ คำสั่งหลายคำสั่งที่เขียนต่อกันเป็นชุด และทำให้เป็นกลุ่มคำสั่งด้วยการครอบด้วย **วงเล็บปีกกา (braces)**

E.g.

```
if (2 < 3)
{
    number = number + 1;
    printf("%d\n", number);
}
```

คำสั่งว่าง (an empty statement) คือ ไม่เขียนอะไร แค่ใส่ **semicolon** ปิดไปเฉยๆ

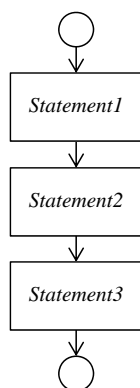
E.g.

```
if (condition);
```

Cited from slides provided by Worawan Worawan Diaz Carballo

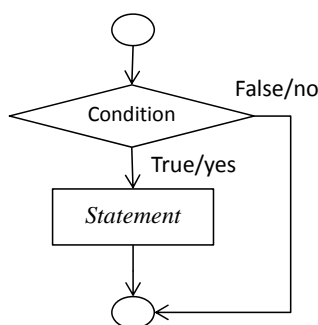
↑

LAST LECTURE



Sequential Structure

- **Sequential** : การทำงานตามลำดับขั้นโดยไม่ข้ามหรือกระโดดไปไหน
- **Decision** : การตัดสินใจทำคำสั่งตาม ผลการเทียบเงื่อนไข หาก เงื่อนไขเป็นจริง จะทำงานที่กำหนดไว้

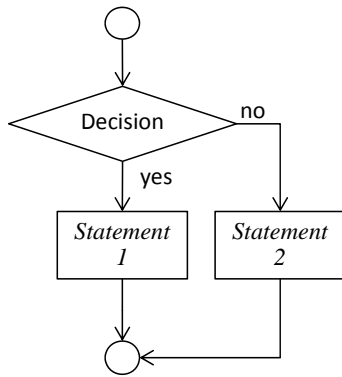


If Statement

C Code If Statement

```
if ( Expression )
{
    statement
}
```

LAST LECTURE



If-Else Statement

C Code If-Else Statement

```

if ( expression )
{
    statement1
}
else
{
    statement2
}
  
```

Statement หรือ Block ที่ตาม
else มา จะทำก็ต่อเมื่อผลการเทียบ
 เงื่อนไขของ **Boolean Expression**
 เป็นเท็จ

C Code If-Else Statement แบบย่อ

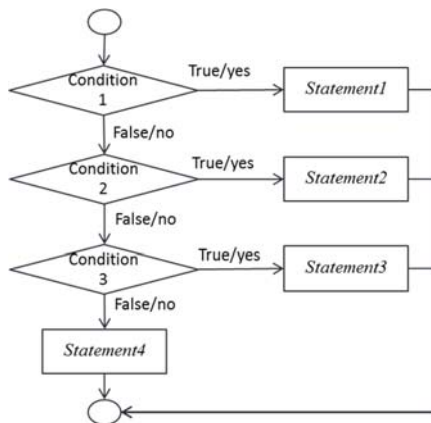
```

expression ? statement1 : statement2
  
```

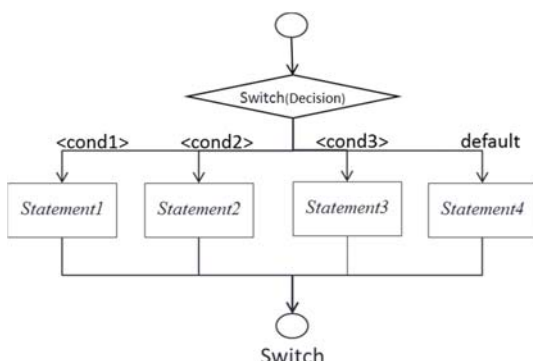
1. $a = (2 > 3) ? 2 : 3$ เงื่อนไข $2 > 3$ เป็นเท็จ ตัวแปร a มีค่าเท่ากับ 3
2. $b = (3 > 2) ? 2 : 3$ เงื่อนไข $3 > 2$ เป็นเท็จ ตัวแปร b มีค่าเท่ากับ 2

LAST LECTURE

C Code Cascaded If Statement



Cascaded if Statement



Switch

```

if ( expression1 )
{
    statement1
}
else if ( expression2 )
{
    statement2
}
else if ( expression3 )
{
    statement3
}
else
{
    statement4
}
  
```

C Code Switch

```

switch (Decision)
{
    case <cond1>:
        statement1; break;
    case <cond2>:
        statement2; break;
    case <cond3>:
        statement3; break;
    default:
        statement4;
};
  
```

THIS LECTURE

• Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
 - Statement and Logical Expression
 - If Statement
 - If-Else Statement
 - Cascaded if statement and Switch (ทบทวน)
 - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

Provided by Natsuda Kaothanthong (1/57)

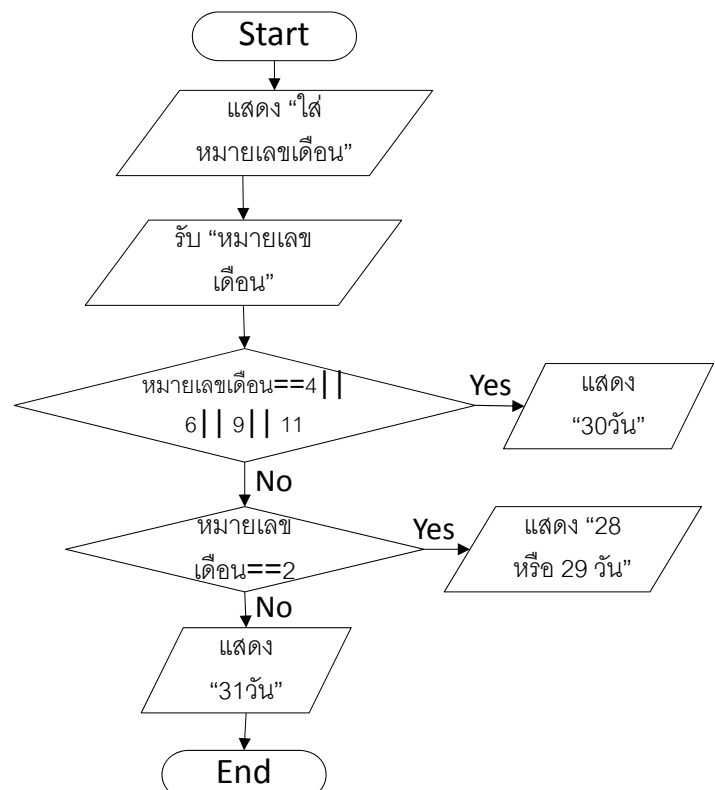
5

ทบทวน CASCADED IF-STATEMENT EXAMPLE: MONTHS.C

จงเขียนโปรแกรมที่รับหมายเลขเดือนเข้ามา
และบอกว่าเดือนนั้นมีกี่วัน

พิจารณาเงื่อนไข

1. เดือนที่มี 30 วันคือ April, June, September, และ November.
2. เดือนอื่นๆ มี 31 วัน
3. ยกเว้นเดือน กุมภาพันธ์ เดือนเดียวที่ส่วนใหญ่แล้วมี 28 วัน หากเป็นปี Leap year จะมีจำนวนวันเท่ากับ 29 วัน

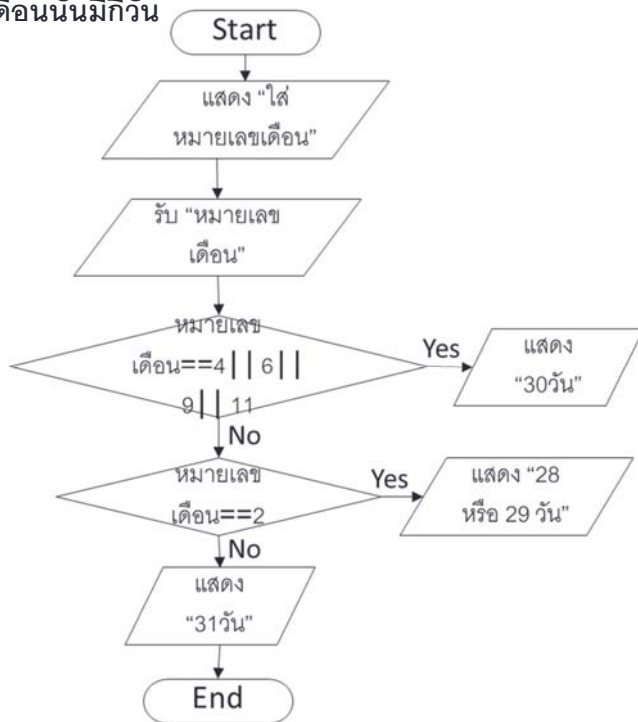


Provided by Natsuda Kaothanthong (1/57)

6

EXAMPLE: MONTHS.C

จงเขียนโปรแกรมที่รับหมายเลขเดือนเข้ามา และบอกว่าเดือนนั้นมีกี่วัน



output "Enter an integer"
input month

if (month is September,
or April,
or June,
or November)

then

{
output "30 days"

}
else if (month is February)

{
output "28 or 29 days"

}
else

{
output "31 days"

}

Provided by Natsuda Kaothanthong (1/57)

7

Example: months.c

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

```
/******\
```

Determine the number of days
in a given month:

คือเวิร์ด **const** ใช้สำหรับประกาศตัวแปรโดย
ค่าของตัวแปรไม่สามารถเปลี่ยนได้ภายหลัง
const มาจากคำว่า **constant**

```
const int September = 9;  
const int April = 4;  
const int June = 6;  
const int November = 11;  
const int February = 2;
```

```
int main()  
{
```

```
return 0;
```

```
}
```

Provided by Natsuda Kaothanthong (1/57)

8

Example: months.c

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

```
/******\
```

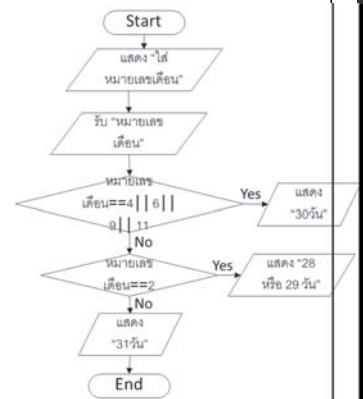
Determine the number of days
in a given month:

```
30 days hath September,  
April, June and November;  
All the rest have 31,  
Excepting February alone,  
And that has 28 days clear  
And 29 in each leap year.  
\\*****/
```

```
const int September = 9;  
const int April = 4;  
const int June = 6;  
const int November = 11;  
const int February = 2;
```

Provided by Natsuda Kaothanthong (1/57)

```
int main()  
{  
    int month;  
    printf("Enter number of month: ");  
    scanf("%d", &month);
```



```
        return 0;
```

```
    }
```

Example: months.c

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

```
/******\
```

Determine the number of days
in a given month:

```
30 days hath September,  
April, June and November;  
All the rest have 31,  
Excepting February alone,  
And that has 28 days clear  
And 29 in each leap year.  
\\*****/
```

```
const int September = 9;  
const int April = 4;  
const int June = 6;  
const int November = 11;  
const int February = 2;
```

Provided by Natsuda Kaothanthong (1/57)

```
int main()  
{  
    int month;  
    printf("Enter number of month: ");  
    scanf("%d", &month);
```

```
    if (month==September ||  
        month==April ||  
        month==June ||  
        month==November )
```

```
    {  
        printf("30 days\n");  
    }
```

```
    else if (month==February)
```

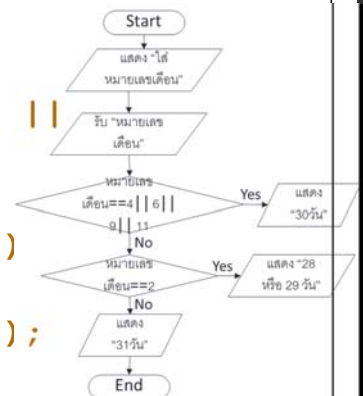
```
    {  
        printf("28 or 29 days\n");  
    }
```

```
    else
```

```
    {  
        printf("31 days\n");  
    }
```

```
    return 0;
```

```
}
```



SWITCH

ให้ลองเขียนFlowchartของโปรแกรม month.c ให้เป็น switch

จงเขียนโปรแกรมที่รับหมายเลขเดือนเข้ามา และ
บอกว่าเดือนนั้นมีกี่วัน

พิจารณาเงื่อนไข

1. เดือนที่มี 30 วันคือ September, April, June และNovember.
2. เดือนอื่นๆ มี 31 วัน
3. ยกเว้นเดือน กุมภาพันธ์ เดือนเดียวที่ส่วนใหญ่แล้วมี 28 วัน หากเป็นปี Leap year จะมีจำนวนวันเท่ากับ 29 วัน

CASCADED IF-STATEMENT

เขียนFlowchartสำหรับโปรแกรมที่รับจำนวนนาฬิกาของการต้มบะหมี่กึ่งสำเร็จรูป แสดงคำว่า “สุกพอดี” , “เส้นแข็ง” และ “เส้นอืด” เมื่อต้มด้วยเวลา 2 นาที, 1 นาที, 3 นาทีตามลำดับ หากต้มนานกว่านั้นให้แสดง “ทำไมอร่อยกว่า” ให้แสดงFlowchartเป็น cascaded if-statement

THIS LECTURE

• Introduction to Control Structures โครงสร้างโปรแกรม

- Sequential Structure การทำงานแบบตามลำดับ
- Decision Structure การตัดสินใจทำตามเงื่อนไข
 - Statement and Logical Expression
 - If Statement
 - If-Else Statement
 - Cascaded if statement and Switch
 - Nested If
- Loop Structure การทำซ้ำ
- Case Structure การทำงานตามกรณี

Provided by Natsuda Kaothanthong (1/57)

13

NESTED IF STATEMENT

- การเขียน If ซ้อน If
- แต่ละblockมีการกำหนดเงื่อนไข หากเงื่อนไขเป็นจริง จะตรวจสอบเงื่อนไขย่อยใน block นั้น
- ถ้าเงื่อนไขย่อยของblockนั้นเป็นจริง statements ของเงื่อนไขนั้น จะถูกกระทำ

แสดงข้อความ"คุณผ่าน"หากได้คะแนนมากกว่า60คะแนน และแสดงเกรด "A"ถ้าได้คะแนนมากกว่า80 "B"ถ้าได้มากกว่า70 มิฉะนั้นแสดง "C" และแสดง "คุณต้องรีเกรด" หากได้คะแนนต่ำกว่า 60คะแนน

In Writing-Cascaded If

```
ถ้า (คะแนนมากกว่า 80)
then แสดง "คุณผ่าน" และ แสดง "A"
มิฉะนั้น ถ้า(คะแนนมากกว่า 70 )
then แสดง "คุณผ่าน" และ แสดง "B"
มิฉะนั้น ถ้า(คะแนนมากกว่า 60 )
then แสดง "คุณผ่าน" และ แสดง "C"
มิฉะนั้น แสดง "คุณต้องรีเกรด"
```

In Writing-Nested If

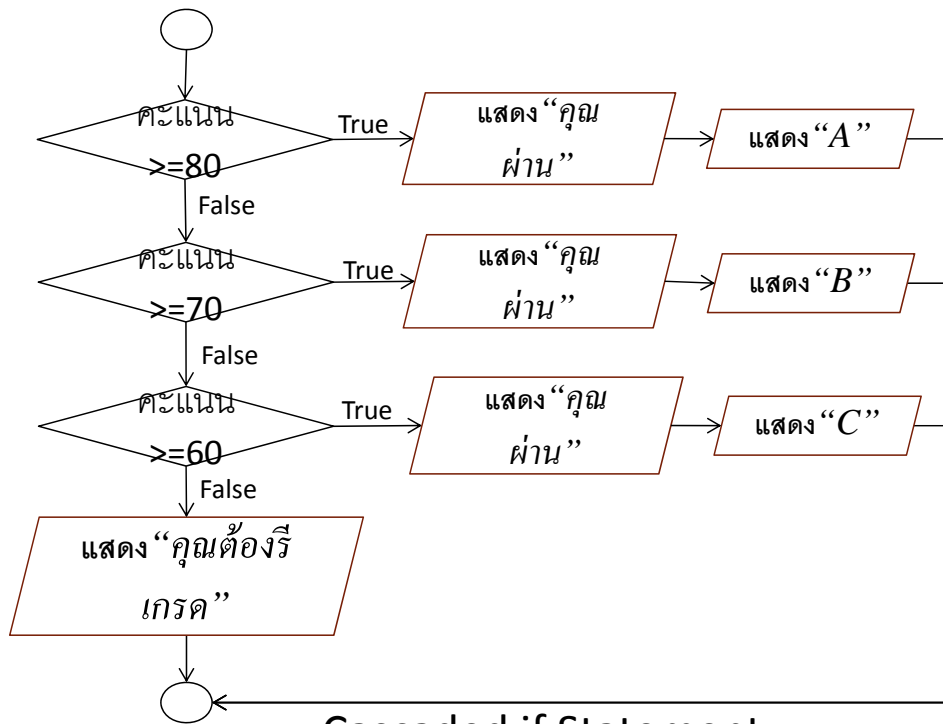
```
ถ้า (คะแนนมากกว่า 60)
then คุณผ่าน
    ถ้า (คะแนนมากกว่า 80)
        ได้เกรด A
    มิฉะนั้นถ้า(คะแนนมากกว่า 70 )
        ได้เกรด B
    มิฉะนั้น
        ได้เกรด C
มิฉะนั้น คุณต้องรีเกรด
```

เงื่อนไขย่อย ของ
ถ้า (คะแนนมากกว่า 60)

Provided by Natsuda Kaothanthong (1/57)

14

CASCADED IF STATEMENT

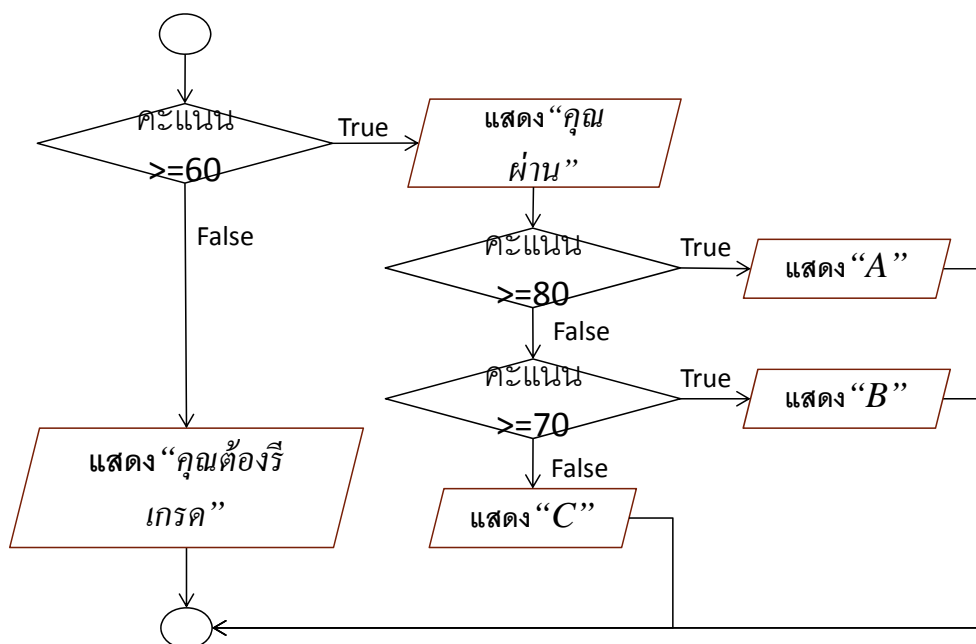


Cascaded if Statement

In Writing-Cascaded If

ถ้า (คะแนนมากกว่า 80)
 then แสดง "คุณผ่าน" และ
 แสดง "A"
 มิฉะนั้น ถ้า(คะแนนมากกว่า 70)
 then แสดง "คุณผ่าน" และ
 แสดง "B"
 มิฉะนั้น ถ้า(คะแนนมากกว่า 60)
 then แสดง "คุณผ่าน" และ
 แสดง "C"
 มิฉะนั้น แสดง "คุณต้องรีเกรด"

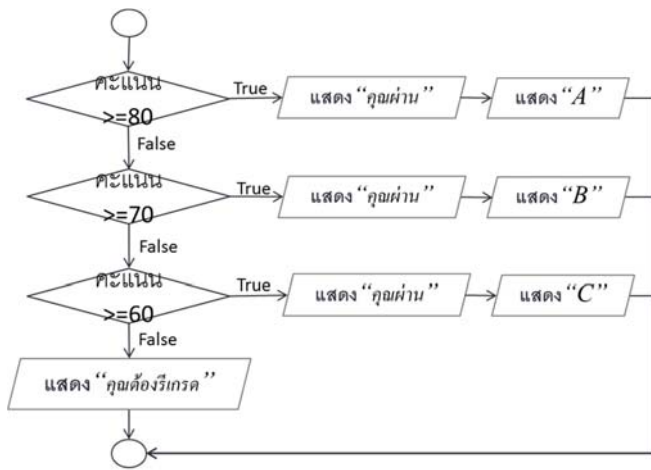
NESTED IF STATEMENT



Nested-if Statement

In Writing-Nested If

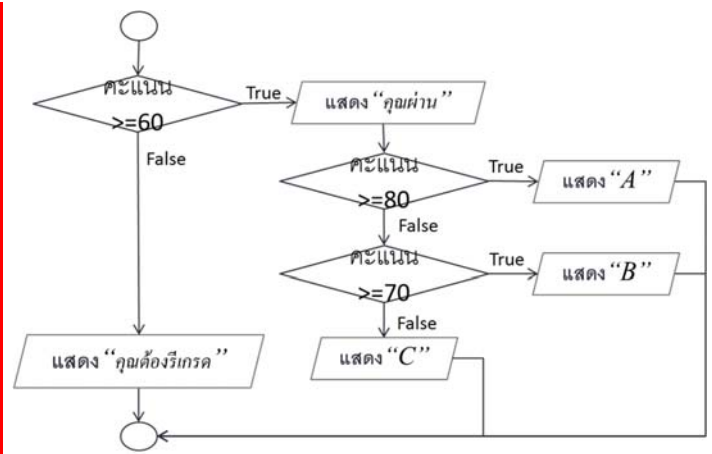
ถ้า (คะแนนมากกว่า 60)
 then คุณผ่าน
 ถ้า (คะแนนมากกว่า 80)
 ได้เกรด A
 มิฉะนั้นถ้า(คะแนนมากกว่า 70)
 ได้เกรด B
 มิฉะนั้น
 ได้เกรด C
 มิฉะนั้น คุณต้องรีเกรด



Cascaded if Statement
In Writing-Cascaded If

ถ้า (คะแนนมากกว่า 80)
then แสดง "คุณผ่าน" และ แสดง "A"
มิฉะนั้น ถ้า (คะแนนมากกว่า 70)
then แสดง "คุณผ่าน" และ แสดง "B"
มิฉะนั้น ถ้า (คะแนนมากกว่า 60)
then แสดง "คุณผ่าน" และ แสดง "C"
มิฉะนั้น แสดง "คุณต้องรีเกร็ด"

Provided by Natsuda Kaothanthong (1/57)



Nested-if Statement
In Writing-Nested If

ถ้า (คะแนนมากกว่า 60)
then คุณผ่าน
 ถ้า (คะแนนมากกว่า 80)
 ได้เกรด A
 มิฉะนั้นถ้า (คะแนนมากกว่า 70)
 ได้เกรด B
 มิฉะนั้น
 ได้เกรด C
มิฉะนั้น คุณต้องรีเกร็ด

17

EXAMPLE OF NESTED IF STRUCTURE

ให้เขียนโปรแกรมที่รับเลขจำนวนเต็ม 3 จำนวนจากผู้ใช้ แล้วหาเลขที่มากที่สุดที่ผู้ใช้ใส่ค่ามา เช่น

6 2 10

สิ่งที่ต้องพิจารณา: เครื่องหมาย <, <=, >, และ >= ใช้เปรียบเทียบระหว่างค่า 2 ค่า เช่น $a < b$ เป็นต้น

กำหนดให้ ค่าแรกที่รับเข้ามาเป็นตัวแปร $n1$, ค่าที่ 2 เป็น $n2$, และ ค่าที่ 3 เป็น $n3$ (int)

ตัวเลขที่มากที่สุด ถูกเก็บในตัวแปร largest (int)

1. ถ้า $n1 > n2$ เป็นจริงต้องพิจารณาด้วยว่า $n1 > n3$
หรือไม่ ถ้าเป็นจริง $largest = n1$ แต่ถ้าเป็นเท็จ
 $largest = n3$
2. แต่ถ้า $n1 > n2$ เป็นเท็จ ต้องพิจารณาด้วยว่า $n2 > n3$
หรือไม่ ถ้าเป็นจริง $largest = n2$ แต่ถ้าเป็นเท็จ
 $largest = n3$

In Writing-Nested If

ถ้า ($n1 > n2$) then
 ถ้า ($n1 > n3$)
 largest = n1
 มิฉะนั้น
 largest = n3
มิฉะนั้น
 ถ้า ($n2 > n3$)
 largest = n2
 มิฉะนั้น
 largest = n3

Provided by Natsuda Kaothanthong (1/57)

18

EXAMPLE OF NESTED IF STRUCTURE 1

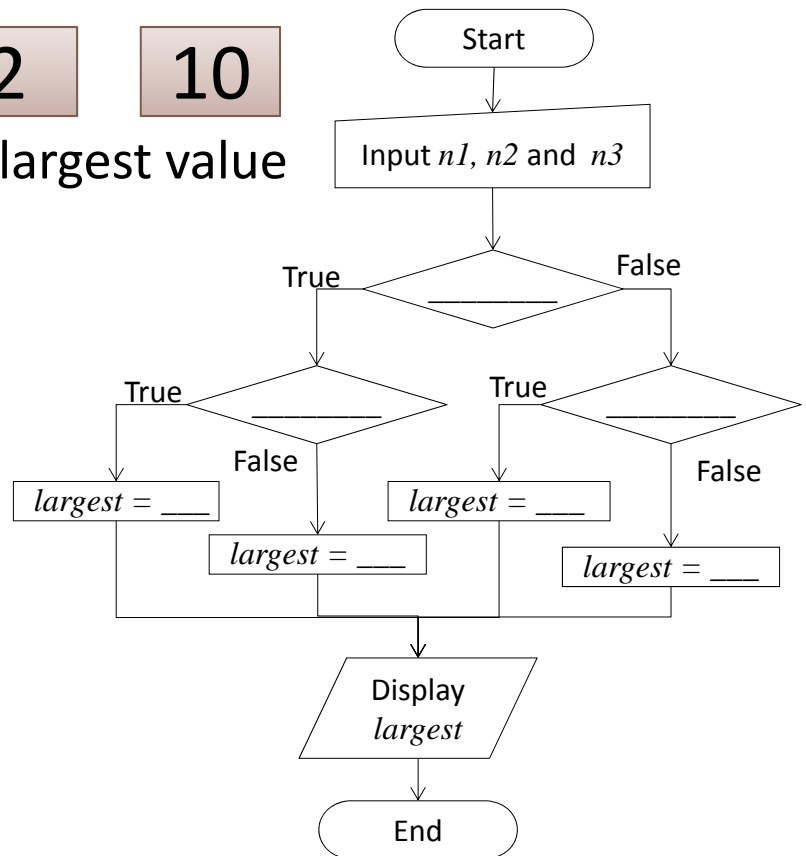
6 2 10

Finding the largest value

In Writing-Nested If

```

ถ้า (n1>n2) then
    ถ้า (n1>n3)
        largest=n1
    มิฉะนั้น
        largest=n3
มิฉะนั้น
    ถ้า (n2>n3)
        largest=n2
    มิฉะนั้น
        largest=n3
    
```

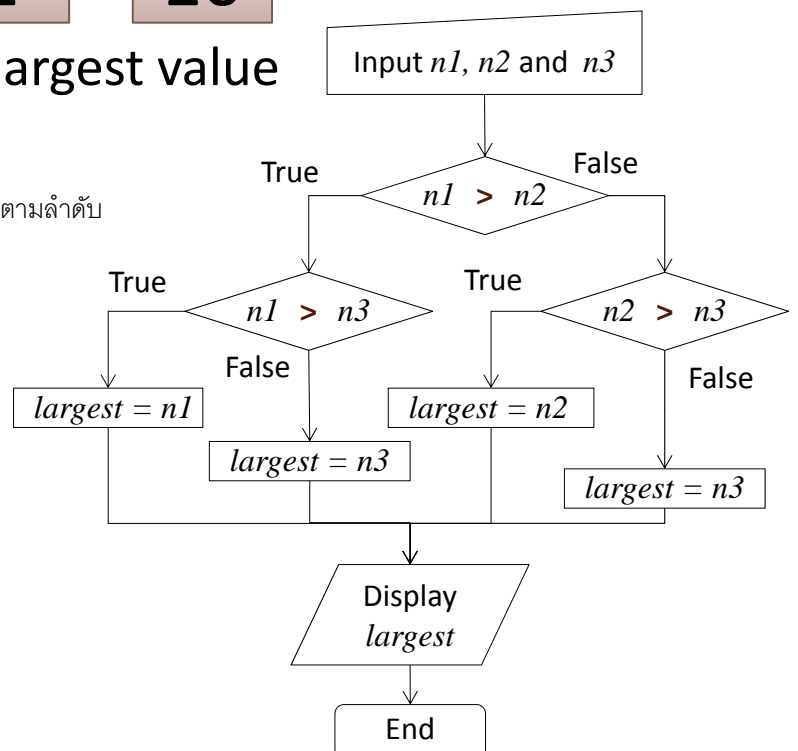


EXAMPLE OF NESTED IF STRUCTURE

6 2 10

Finding the largest value

- กำหนดให้
 - largest* เป็นค่าที่มากที่สุด
 - n1*, *n2* และ *n3* เป็นตัวเลข 3 ตัวที่รับเข้ามาตามลำดับ
- รับเลขจำนวนเต็ม 3 ตัวจากผู้ใช้
- ถ้า $n1 > n2$
 - ✓ ถ้า $n1 > n3$
 - ✓ $largest = n1$
 - × $largest = n3$
 - × ถ้า $n2 > n3$
 - ✓ $largest = n2$
 - × $largest = n3$
- Display *largest*



```

#include <stdio.h>
int main()
{
    int n1, n2, n3; // Input variables
    int largest; // Largest value

    printf("Enter 3 values:");
    scanf("_____", _____);

    if ( _____ )
    {
        if ( _____ )
        { _____ }
        else
        { _____ }
    }
    else
    {
        if ( _____ )
        { _____ }
        else
        { _____ }
    }
    return 0;
}

```

Provided by Natsuda Kaothanthong (1/57)

EXAMPLE OF NESTED IF STRUCTURE 2

เขียนโปรแกรมที่รับราคาขายและเงินที่ลูกค้าชำระค่าสินค้า โดยมีเงื่อนไขดังนี้

ถ้าจำนวนเงินที่ชำระไม่พอ ให้แสดง "เงินไม่พอชำระค่าสินค้า"

แต่ถ้าจำนวนเงินพอ ให้ทอนเงินให้ลูกค้า โดยชนิดของเงินทอนมี 4 ชนิดคือ ธนบัตรใบละ 20 บาท เหรียญ 10, 5, และ 1 บาท โดยให้ทอนจากหน่วยเงินที่มีค่าใหญ่ที่สุดก่อน

พิจารณากรณีตัวอย่าง

1. สินค้าราคา 18 บาท ลูกค้าชำระเงิน 15 บาท

2. สินค้าราคา 11 บาท ลูกค้าชำระเงิน 50 บาท

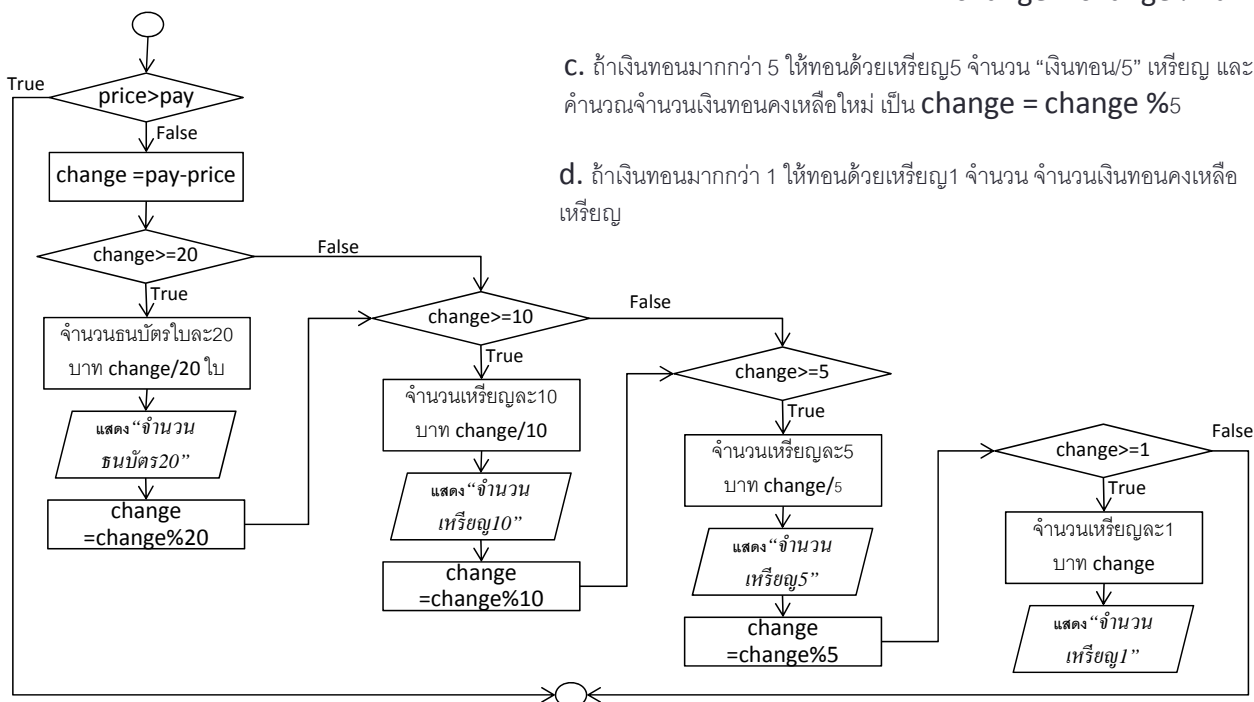
พิจารณาข้อมูล (ชนิดข้อมูล)

1. ราคาสินค้า (_____)
2. จำนวนเงินที่ชำระ (_____)
3. เงินทอน (_____)

เงินทอน เป็น `change (int)`, ราคาสินค้า เป็น `price (int)`, จำนวนเงินที่ชำระ เป็น `pay (int)`

พิจารณาเงื่อนไข

1. ถ้าเงินไม่พอชำระ แสดง "เงินไม่พอ"
2. มิฉะนั้น



Provided by Natsuda Kaothanthong (1/57)

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int change, price, pay;
6      int num_20, num_10, num_5, num_1;
7
8      printf("Enter price:");
9      scanf("%d", &price);
10
11     printf("Enter money paid:");
12     scanf("%d", &pay);
13
14     if (price > pay)
15     {
16         printf("Money not enough!\n");
17     }
18     else
19     {
20         change = pay - price; // Compute change
21         printf("Change is %d Baht\n", change);
22         if (change >= 20)
23         {
24             num_20 = change / 20; // Number of 20 Baht notes
25             printf("20 Baht\t:\t%d Notes\n", num_20);
26             change = change % 20; // Left over change
27         }
28         if (change >= 10)
29         {
30             num_10 = change / 10; // Number of 10 Baht coins
31             printf("10 Baht\t:\t%d Coins\n", num_10);
32             change = change % 10; // Left over change
33         }
34         if (change >= 5)
35         {
36             num_5 = change / 5; // Number of 5 Baht coins
37             printf("5 Baht\t:\t%d Coins\n", num_5);
38             change = change % 5; // Left over change
39         }
40         if (change >= 1)
41         {
42             printf("1 Baht\t:\t%d Coins\n", change); // Number of 1 Baht coins
43         }
44     }
45     return 0;
46 }
  
```

