

w04-Lab

Conditional Control Statement

Part I

Adapted for 204111
by Areerat Trongratsameethong

Flow of Control

- **Flow of Control:** refers to the order in which a program's statements are executed
- **Flow of Control:** คือ ลำดับของของการดำเนินการคำสั่งต่าง ๆ ในโปรแกรม
- Any algorithm can be built using combinations of four standardized flow of control structures:
- Algorithm ใด ๆ สามารถสร้างให้อยู่ในมาตรฐานของโครงสร้างการควบคุม ใน 4 รูปแบบ ดังต่อไปนี้
 - Sequential
 - Selection
 - Repetition
 - Invocation

Topics

- **Flow of Control**
- **Relational Expression**
- **The One-way if**
- **The if-else Statements**
- **The Nested if Statements**

Flow of Control (2)

- **Sequential:** Normal flow of control for all programs คือ การดำเนินการคำสั่งตามลำดับตลอดทั้งโปรแกรม
- **Selection:** is used to select which statements are performed next based on a condition คือ การดำเนินการคำสั่งตามเงื่อนไข
- **Repetition:** is used to repeat a set of statements คือ การดำเนินการชุดคำสั่งซ้ำ
- **Invocation:** is used to invoke a sequence of instructions using a single statement, as in calling a function คือ การดำเนินการชุดของคำสั่งที่รวมอยู่ด้วยกันซึ่งอยู่ในรูปแบบของฟังก์ชัน โดยการเรียกใช้ฟังก์ชันดังกล่าว

Topics

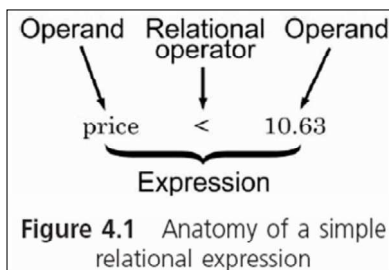
- Flow of Control
- Relational Expression
- The One-way if
- The if-else Statements
- The Nested if Statements

Relational Expression (2): Relational Operators

Table 4.1 Relational Operators in C

Relational Operator	Meaning	Example
<	less than	age < 30
>	greater than	height > 6.2
<=	less than or equal to	taxable <= 20000
>=	greater than or equal to	temp >= 98.6
==	equal to	grade == 100
!=	not equal to	number != 250

Example



Relational Expression

- Relational expressions are also known as conditions
- A relational expression evaluates to 1 (true) or 0 (false)
 - The expression 3 < 4 has a value of 1 (เป็นจริง)
 - The expression 2.0 > 3.3 has a value of 0 (เป็นเท็จ)
 - The value of hours > 0 depends on the value of hours ในกรณี hours > 0 ผลลัพธ์ขึ้นอยู่กับค่า hours

Relational Expression (3)

- Character data can also be compared using relational operators

Table 4.2 Sample Comparisons of ASCII Characters

Expression	Value	Interpretation
'A' > 'C'	0	false
'D' <= 'Z'	1	true
'E' == 'F'	0	false
'g' >= 'm'	0	false
'b' != 'c'	1	true
'a' == 'A'	0	false
'B' < 'a'	1	true
'b' > 'Z'	1	true

Topics

- Flow of Control
- Relational Expression
- The One-way if
- The if-else Statements
- The Nested if Statements

One-way if (2)



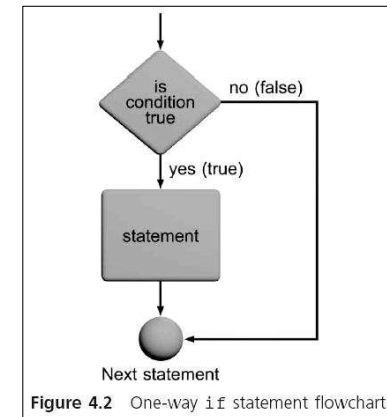
Program 4.1

```

1 #define LIMIT 3000.0
2 #include <stdio.h>
3
4 int main()
5 {
6     int idNum;
7     float miles;
8
9     printf("Please type in car number and mileage: ");
10    scanf("%d %f", &idNum, &miles);
11
12    if(miles > LIMIT) ← No semicolon (;) here
13        printf(" Car %d is over the limit.\n",idNum); } One-way if statement
14
15    printf("End of program output.\n");
16
17    return 0;
18 }
```

One-way if

- Simplest decision structure:
เงื่อนไขที่ดำเนินการทางเดียว
if (expression)



statement executed if condition is true

- เงื่อนไข (expression) จะถูกตรวจสอบว่าเป็นจริงหรือเท็จ
- ถ้าผลลัพธ์จากการตรวจสอบเป็นจริง คำสั่งที่อยู่ต่อจาก if จะถูกดำเนินการ แต่ถ้าผลลัพธ์จากการตรวจสอบเป็นเท็จ จะไม่มีการดำเนินการใดๆ

Compound Statement

- คำสั่งที่ให้ดำเนินการหลังจากตรวจสอบเงื่อนไข มี 2 ลักษณะ คือ

- One Statement: ดำเนินการเพียง 1 คำสั่ง

```
if (expression)
{
    Statement1;
}
```

```
if (expression)
    Statement1;
```

สามารถละ () ได้ กรณีเป็น one statement

- Compound Statements: ดำเนินการหลายคำสั่ง

```
if (expression)
{ // Open Block must be put
    Statement1;
    Statement2;
    ...
    Statementn;
} // End Block must be put
```

Example:

```
if (x < 10)
{
    printf("x = %d\n", x);
    x = x + 10;
    printf("x + 10 = %d\n", x);
}
```

Topics

- Flow of Control
- Relational Expression
- The One-way if
- The if-else Statements
- The Nested if Statements



Program 4.2

The if-else Statement (2)

```

1 #include <stdio.h>
2 #define LOWRATE 0.02 /* lower tax rate */
3 #define HIGHRATE 0.025 /* higher tax rate */
4 #define CUTOFF 20000.0 /* cut off for low rate */
5 #define FIXEDAMT 400 /* fixed dollar amount for higher rate amounts */
6
7 int main()
8 {
9     float taxable, taxes;
10
11     printf("Please type in the taxable income: ");
12     scanf("%f", &taxable);
13
14     if (taxable <= CUTOFF)
15         taxes = LOWRATE * taxable;
16     else
17         taxes = HIGHRATE * (taxable - CUTOFF) + FIXEDAMT;
18
19     printf("Taxes are $%7.2f\n", taxes);
20
21     return 0;
22 }
```

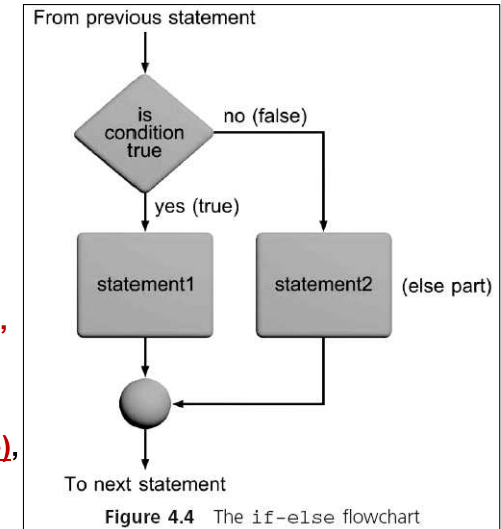
The if-else Statement

- The most commonly used if-else statement is:

```

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

- If the value of *expression* is **1 (true)**, the *statement1* is executed
- If the value of *expression* is **0 (false)**, the *statement2* is executed



Program 4.3

The if-else Statement (3)

```

1 #include <stdio.h>
2
3 int main()
4 {
5     char tempType;
6     float temp, fahrenheit, celsius;
7
8     printf("Enter the temperature to be converted: ");
9     scanf("%f", &temp);
10    printf("Enter an f if the temperature is in Fahrenheit\n");
11    printf(" or a c if the temperature is in Celsius: ");
12    scanf("\n%c", &tempType); /* see footnote on page 111 */
13
14    if (tempType == 'f')
15    {
16        celsius = (5.0 / 9.0) * (temp - 32.0);
17        printf("\nThe equivalent Celsius temperature is %6.2f\n", celsius);
18    }
19    else
20    {
21        fahrenheit = (9.0 / 5.0) * temp + 32.0;
22        printf("\nThe equivalent Fahrenheit temperature is %6.2f\n", fahrenheit);
23    }
24
25    return 0;
26 }
```

Topics

- Flow of Control
- Relational Expression
- The One-way if
- The if-else Statements
- **The Nested if Statements**

The Nested if Statements

- Nested if statement:

```

if (expression1)
{
    statement1;
}
else
{
    if (expression2)
    {
        statement2;
    }
    else
    {
        statement3;
    }
}

```

- If the value of *expression1* is true, the *statement1* is executed
- Otherwise (the *expression1* is false)
 - If the value of *expression2* is true, the *statement2* is executed
 - Otherwise (the *expression1* and *expression2* are false) the *statement3* is executed

The Nested if Statements

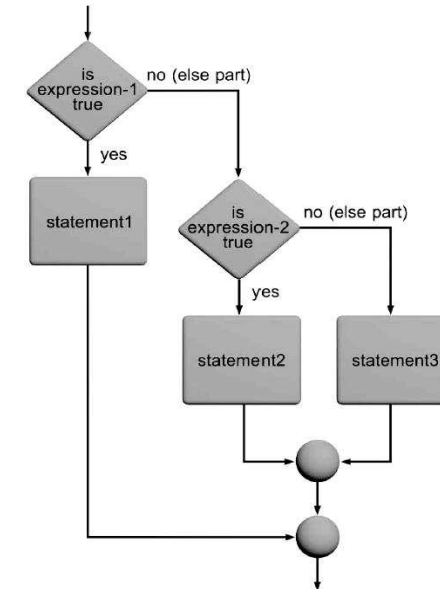


Figure 4.5 The if-else chain's flowchart

The Nested if Statement (2)

- สิ่งที่ต้องระวังสำหรับการใช้ Nested if
 - Compiler จะจับคู่ else กับ if ที่อยู่ก่อนหน้าที่ใกล้ที่สุดเสมอ ถึงแม้จะมีการใส่เยื้องเพื่อบอกกลุ่มของ if ดังตัวอย่างแสดงด้านล่าง

- Nested if statement:

```

if (x > 0)
{
    if (x > 100)
        printf("x is greater than 100");
    else
        if (x < -100)
            printf("x is less than -100");
        else
            printf("x is greater than or equal to -100");
}

```

Compiler มองสองตัวนี้คู่กัน

Input (x)	Output
150	?
-50	?
-200	?

The Nested if Statement (3)

- วิธีแก้ปัญหาในการแบ่งกลุ่ม Nested if: ให้ใส่ Block ({ ... }) เพื่อแยกกลุ่มของ if-else แต่ละกลุ่ม ดังตัวอย่างแสดงด้านล่าง

```

if ( x > 0 )
{
    if ( x > 100 )
        printf("x is greater than 100");
}
else
{
    if ( x < -100 ) {
        printf("x is Less than -100");
    }
    else {
        printf("x is greater than or equal to -100");
    }
}
    
```

Input (x)	Output
150	?
-50	?
-200	?

Practice 2: if-else

Coding the below problem

Problem Statement

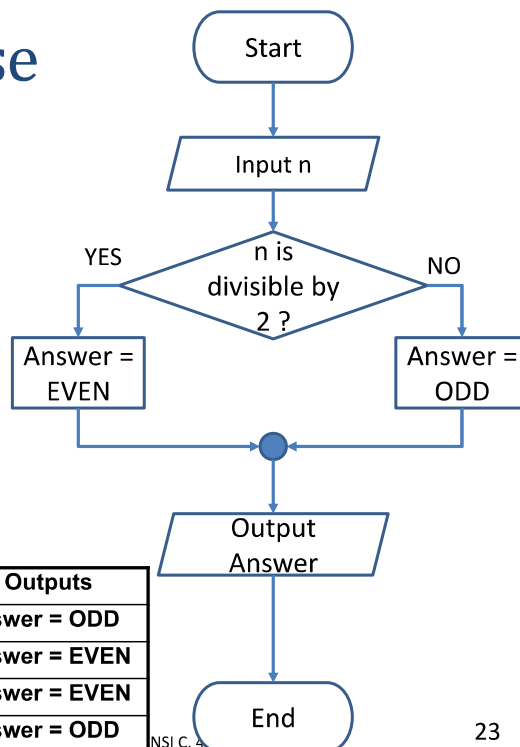
Read a number from the keyboard.
Check and output if a given number N is ODD or EVEN.

Pseudocode

```

read n
if n is divisible by 2 then
    set Answer as EVEN
else
    set Answer as ODD
print Answer
    
```

Inputs	Outputs
N = 5	Answer = ODD
N = 8	Answer = EVEN
N = 0	Answer = EVEN
N = -1	Answer = ODD



Practice 1: One-way if

Coding the below problem

Problem Statement

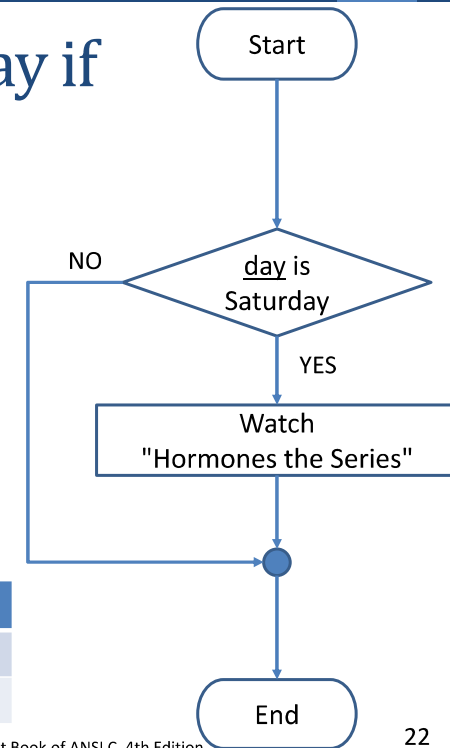
- ถ้าวันนี้เป็นวันเสาร์
- ดู Hormones the Series

Pseudocode

```

if day is Saturday then
    watch Hormones the series
    
```

Expression	Statement
today == "Saturday"	"Watch Hormones the Series"
today != "Saturday"	do nothing

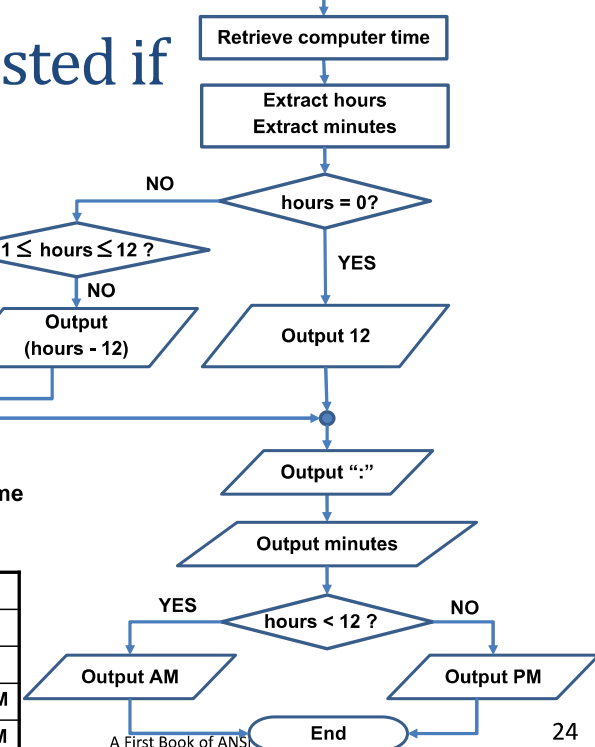


Practice 3: Nested if

Problem Statement

Given computer time is stored in 24 hours format, you are to print the time in AM/PM format

Inputs	Outputs
Computer time = 8:30	Printed time – 8:30 AM
Computer time = 20:30	Printed time – 8:30 PM
Computer time = 0:15	Printed time – 12:15 AM
Computer time = 12:15	Printed time – 12:15 PM



Practice 3: Nested if (2)

Code the below problem

Statement

Given computer time is stored in 24 hours format, you are to print the time in AM/PM format

Inputs	Outputs
Computer time = 8:30	Printed time – 8:30 AM
Computer time = 20:30	Printed time – 8:30 PM
Computer time = 0:15	Printed time – 12:15 AM
Computer time = 12:15	Printed time – 12:15 PM

- Pseudocode

Enter computer time

Extract hours and minutes

```
if hours is equal to 0 then
```

```
  print 12
```

```
else
```

```
  if hours is between* 1 and 12 then
```

```
    print hours
```

```
  else
```

```
    print hours - 12
```

```
print ":"
```

```
print minutes
```

```
if hours is less than 12 then
```

```
  print AM
```

```
else
```

```
  print PM
```