

คำสั่งควบคุม

Conditional Statements

วีระยุทธ คุณรัตนสิริ

เนื้อหา

นิพนธ์ตรรกศาสตร์

คำสั่งแบบมีเงื่อนไข

ตัวอย่าง

โพลีชาร์ต

นิพจน์ตรรกศาสตร์

สัญลักษณ์คณิตศาสตร์	สัญลักษณ์ C#	ตัวอย่าง	ความหมาย
=	==	$x == y$	x เท่ากับ y?
\neq	!=	$x != y$	x ไม่เท่ากับ y?
>	>	$x > y$	x มากกว่า y?
\geq	>=	$x >= y$	x มากกว่าหรือเท่ากับ y?
<	<	$x < y$	x น้อยกว่า y?
\leq	<=	$x <= y$	x น้อยกว่าหรือเท่ากับ y?

ตัวอย่าง: นิพจน์ตรรกศาสตร์

```
double x = 4.0;
```

Expression	Value
------------	-------

<code>x < 5.0</code>	<u><i>true</i></u>
-------------------------	--------------------

<code>x > 5.0</code>	<u><i>false</i></u>
-------------------------	---------------------

<code>x <= 5.0</code>	<u><i>true</i></u>
--------------------------	--------------------

<code>5.0 == x</code>	<u><i>false</i></u>
-----------------------	---------------------

<code>x != 5.0</code>	<u><i>true</i></u>
-----------------------	--------------------

ตัวอย่าง

```
double n1 = 78.0;  
double n2 = 80.0;
```

```
n1 < n2
```

true

```
n1 >= n2
```

false

```
(n1 + 35) > n2
```

true

```
Math.Abs(n1-n2) <= 0.001
```

false

```
n1 == n2
```

false

```
n1 != n2
```

true

การผสมนิพจน์ตรรกศาสตร์

ตัวดำเนินการตรรกศาสตร์

- `&&` - และ (AND)
- `||` - หรือ (OR)
- `!` - ไม่ (NOT)

ตัวอย่าง

- `(x > 12) && (x < 20)`
 - เป็นจริงถ้า `x` มีค่าอยู่ระหว่าง 12 และ 20
 - ถ้า `x` เท่ากับ 10 แล้วนิพจน์ตรรกศาสตร์ข้างต้นเป็นเท็จ
 - ถ้า `x` เท่ากับ 15 แล้วนิพจน์ตรรกศาสตร์ข้างต้นเป็นจริง

ลำดับการประมวลผล (Precedence Rules)

1. () วงเล็บ
2. *, / , %
3. + -
4. <, >, <=, >=
5. ==, !=
6. !
7. &&
8. ||
9. ถ้าลำดับเท่ากันให้ประมวลผลจากซ้ายไปขวา

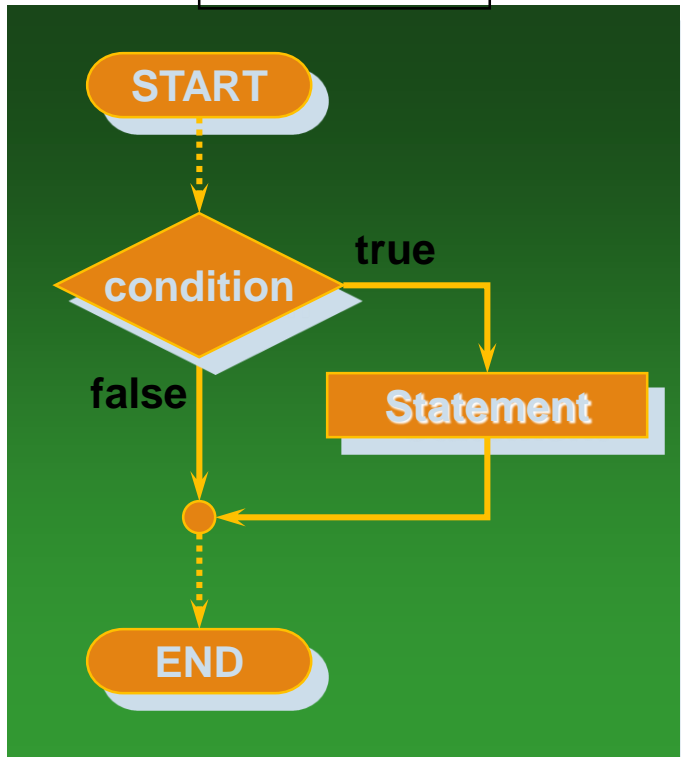
คำสั่งแบบมีเงื่อนไข

if...else

switch...case

คำสั่ง if

Flowchart



C# Syntax

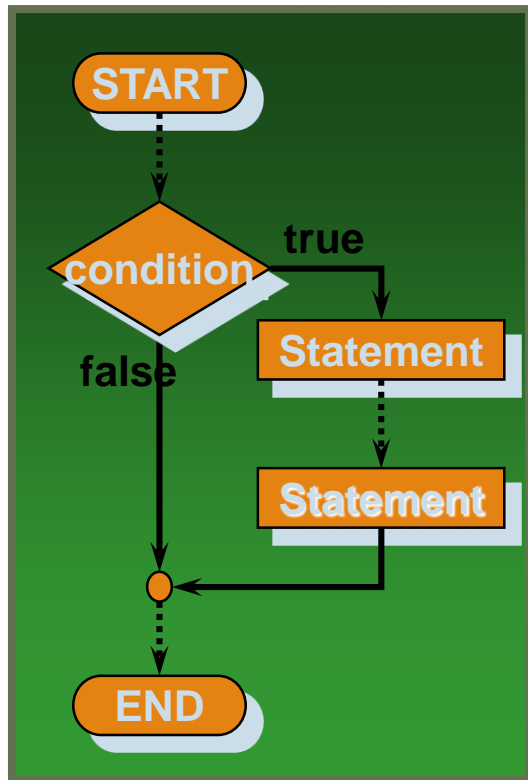
```
if (condition)  
statement;
```

อนุญาตให้มีได้เพียงหนึ่ง
คำสั่งเท่านั้น

การผสมคำสั่ง

เราสามารถสร้างคำสั่งได้มากกว่าหนึ่งคำสั่งโดยการใส่คำสั่งไว้ใน { ... }

Flowchart



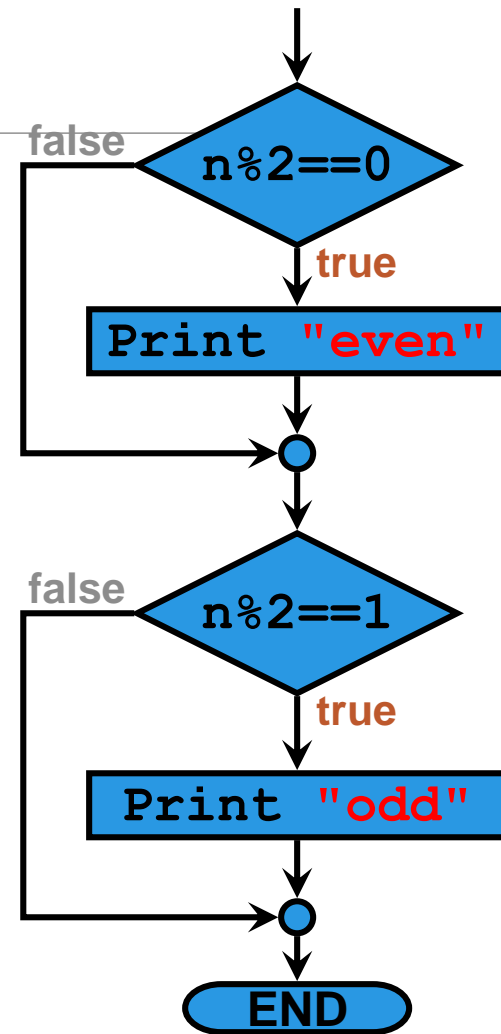
C# Syntax

```
if (condition)  
{  
    statement1;  
    statement2;  
    :  
}
```

ตัวอย่าง: คำสั่ง if

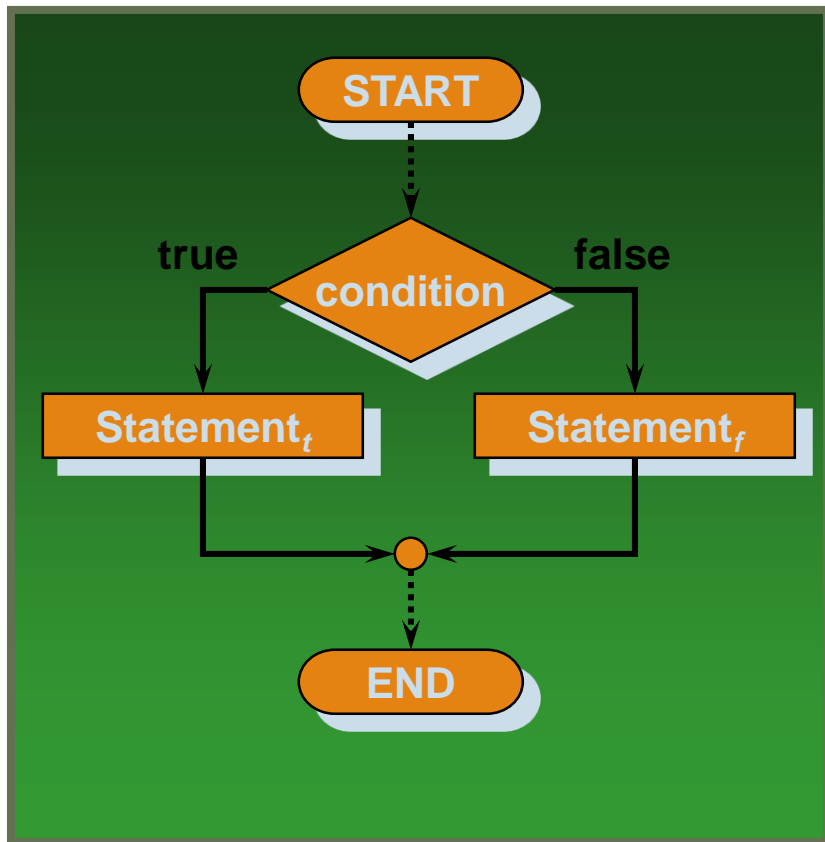
อ่านตัวเลขหนึ่งจำนวน และหาว่าตัวเลขนั้น
เป็นจำนวนคี่ หรือจำนวนคู่

```
using System;  
class EvenOrOdd {  
    static void Main() {  
        int n = int.Parse(Console.ReadLine());  
        if (n % 2 == 0)  
            Console.WriteLine("{0} is even", n);  
        if (n % 2 == 1)  
            Console.WriteLine("{0} is odd", n);  
    }  
}
```



คำสั่ง `if...else...`

Flowchart



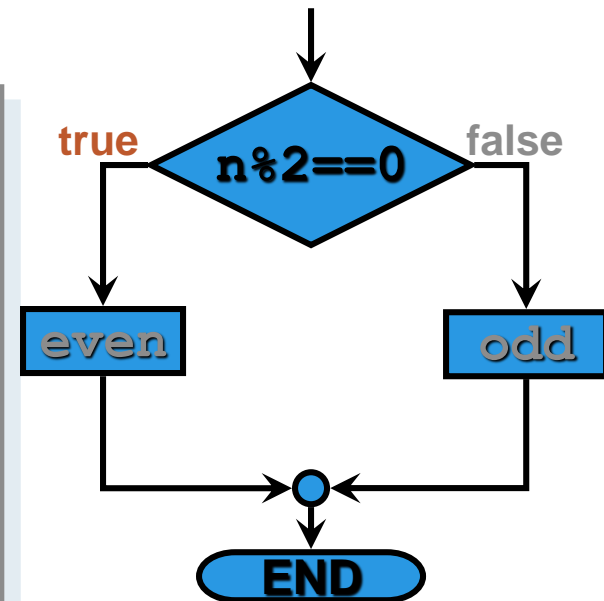
C# Syntax

```
if (condition)  
    statementt;  
else  
    statementf;
```

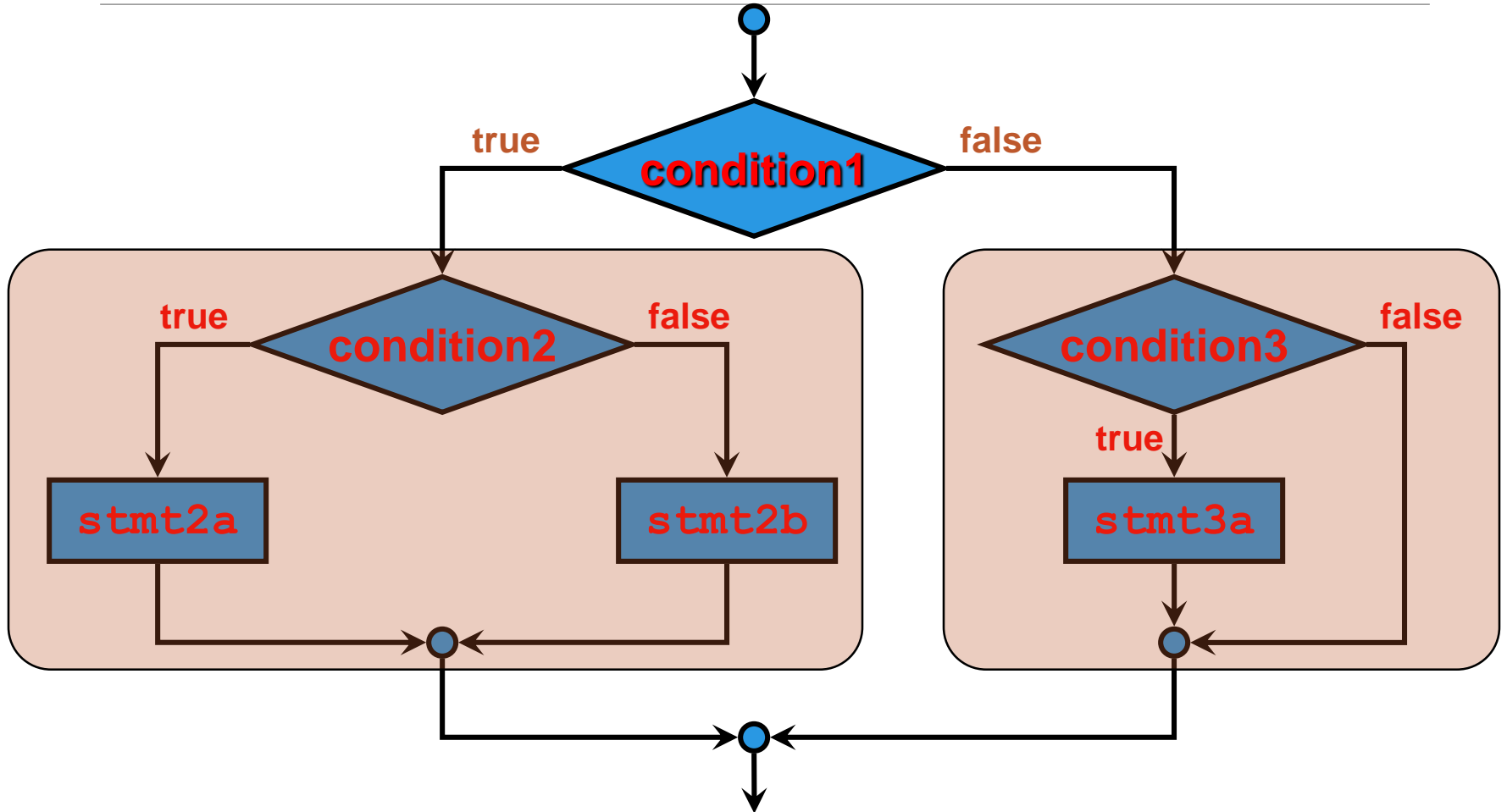
ตัวอย่าง: คำสั่ง `if...else`

อ่านตัวเลขหนึ่งจำนวน และหาว่าตัวเลขนั้นเป็นจำนวนคี่ หรือจำนวนคู่

```
using System;
class EvenOrOdd {
    static void Main() {
        int n = int.Parse(Console.ReadLine());
        if (n % 2 == 0)
            Console.WriteLine("{0} is even", n);
        else
            Console.WriteLine("{0} is odd", n);
    }
}
```

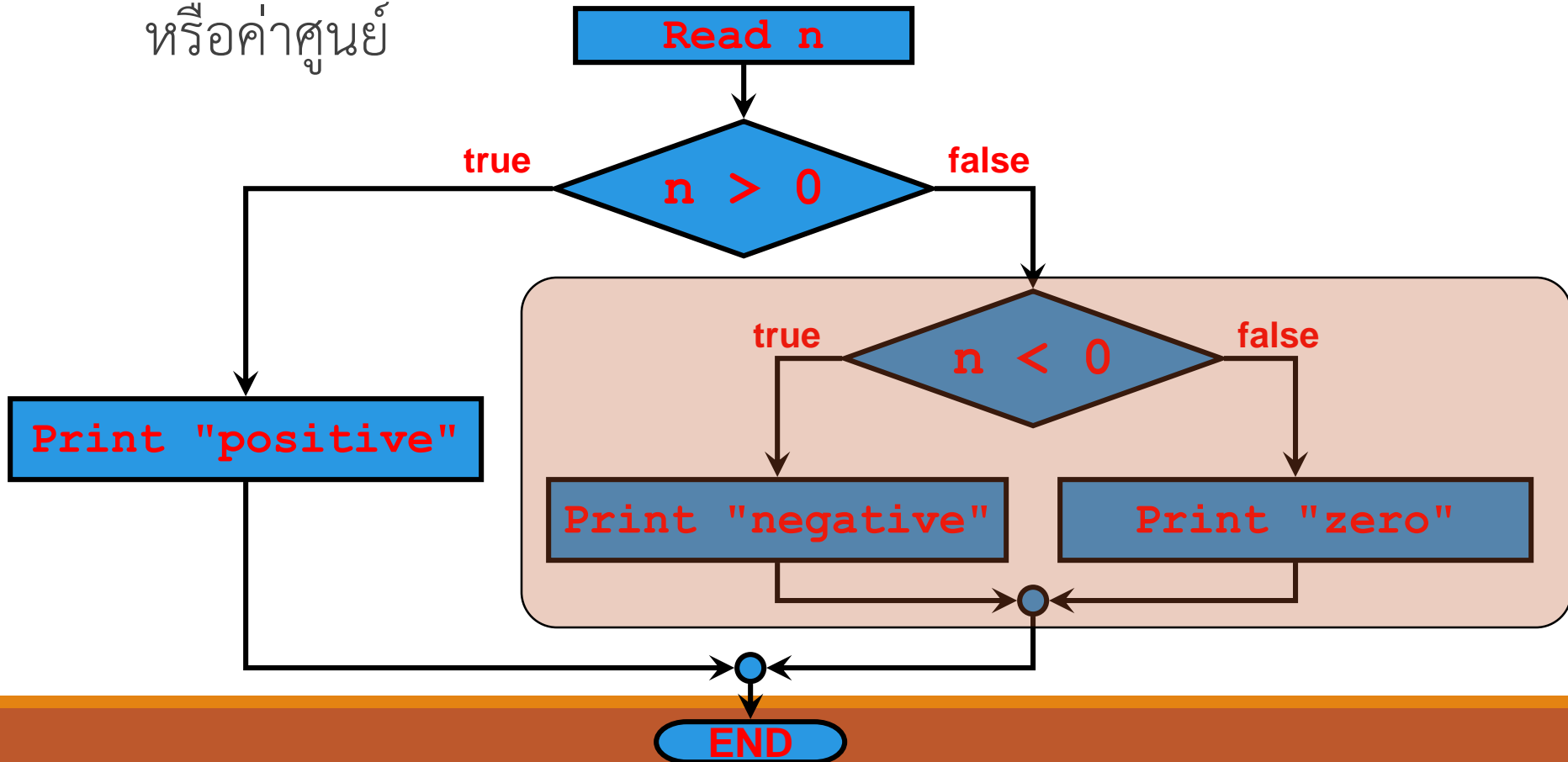


if หลายชั้น



ตัวอย่าง: if หลายชั้น

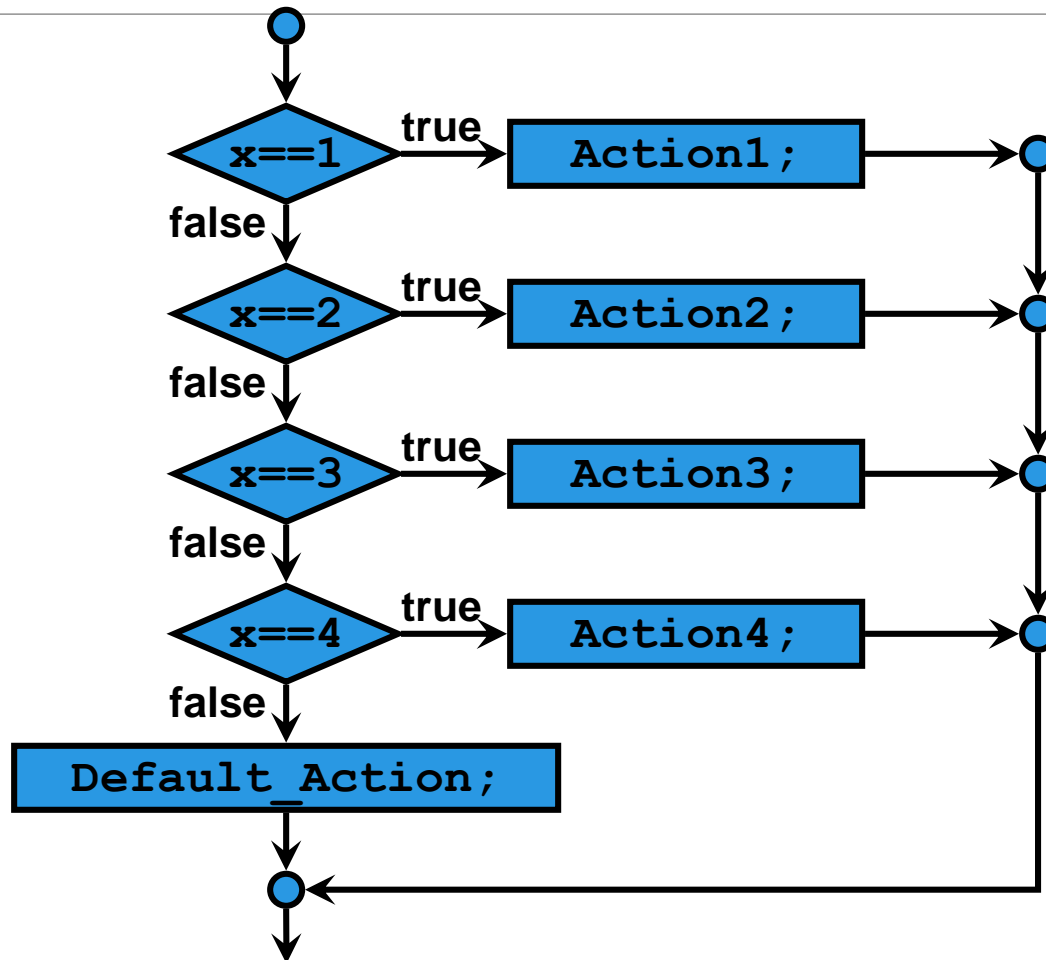
อ่านตัวเลขหนึ่งจำนวน และหาว่าตัวเลขนั้นเป็นค่าบวก ค่าลบ หรือค่าศูนย์



ตัวอย่าง: โปรแกรม C#

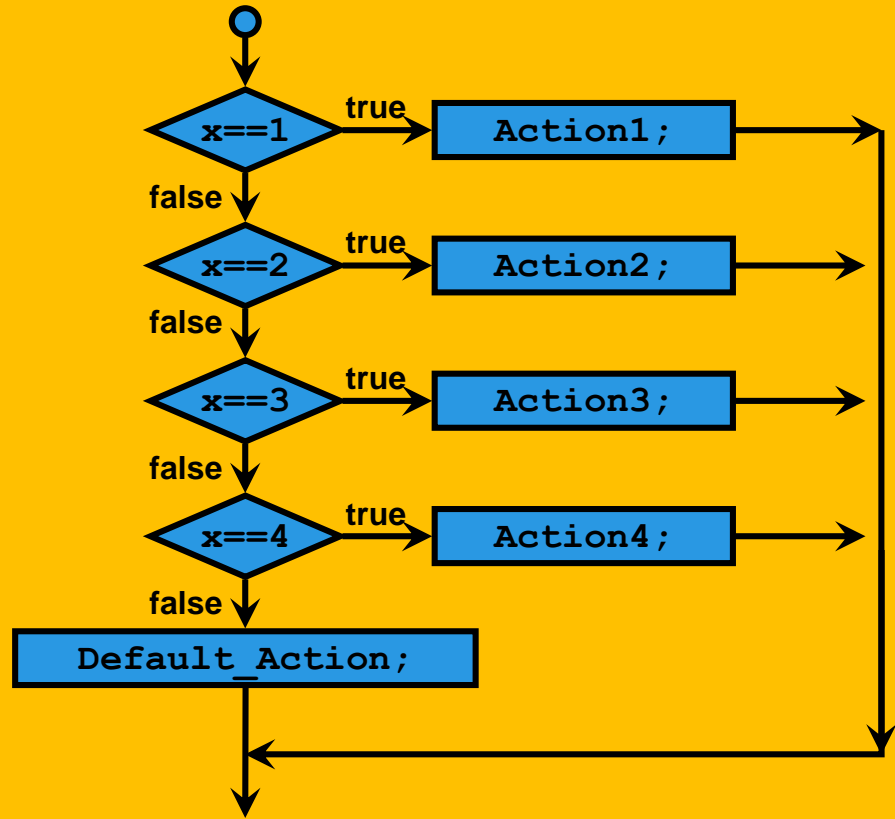
```
using System;
class Sign {
    static void Main() {
        int n = int.Parse(Console.ReadLine());
        if (n > 0)
            Console.WriteLine("{0} is positive", n);
        else {
            if (n < 0)
                Console.WriteLine("{0} is negative", n);
            else
                Console.WriteLine("{0} is zero", n);
        }
    }
}
```


การกำหนดหลายเงื่อนไข



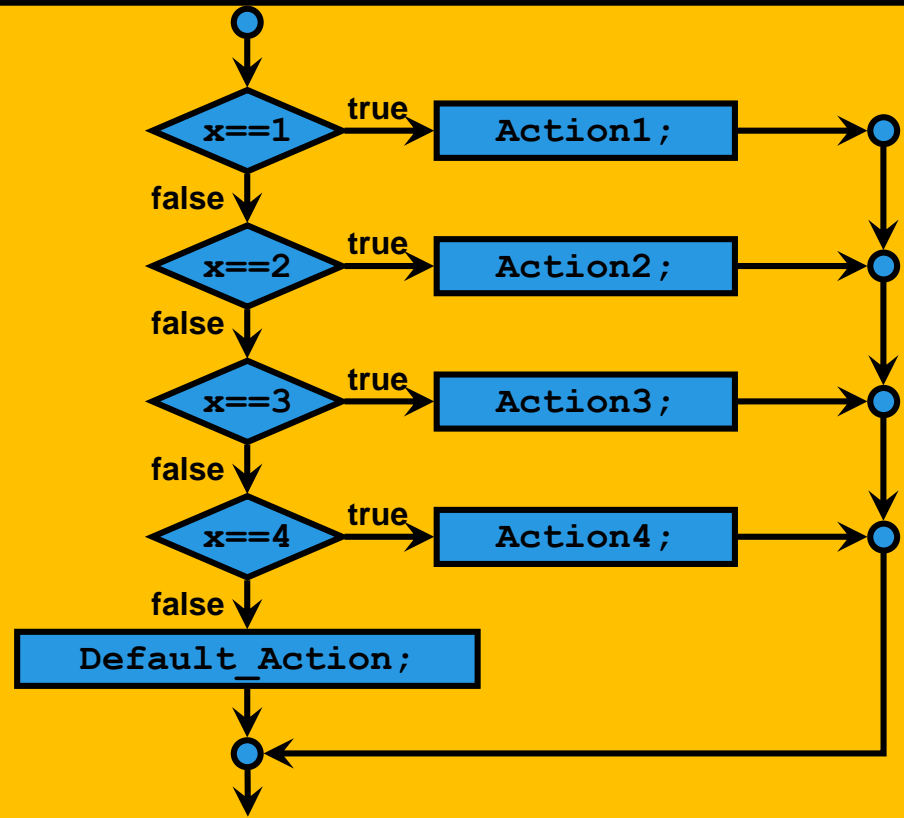
การกำหนดหลายเงื่อนไขกับคำสั่ง if

```
if (x==1)
    Action1;
else if (x==2)
    Action2;
else if (x==3)
    Action3;
else if (x==4)
    Action4;
else
    Default_Action;
```



การใช้คำสั่ง switch...case

```
switch (x) {  
  case 1: Action1;  
    break;  
  case 2: Action2;  
    break;  
  case 3: Action3;  
    break;  
  case 4: Action4;  
    break;  
  default: Default_Action;  
    break;  
}
```



ไวยากรณ์ switch...case

Syntax:

```
switch (<expression>
{
    case <constant-expression>:
        <statements>;
        break;

    default:
        <statements>;
        break;
}
```

<expression> ต้องเป็นประเภทข้อมูล **int, char** หรือ **string**

แบบฝึกหัด

คำนวณค่าใช้จ่ายโทรศัพท์โดยคิดตามเวลาที่ใช้

Operator: LEMON

Promotion	Start Prices (baht)	Free Call (Minutes)	Next Call
SAD	1200	600	2 baht/min
BAD	500	250	5 baht/min

อินพุต: ประเภทโปรโมชั่น (promotion type)
จำนวนเวลาที่ใช้ (usage time)

เอาต์พุต: ค่าใช้จ่าย (payment price)

