

บทที่ 6

การเลือกทำ

ในการเขียนโปรแกรมบางครั้งจะต้องมีการให้โปรแกรมเลือกทำเพื่อที่จะทำงานได้อย่างใดอย่างหนึ่ง โดยถ้าเงื่อนไขเป็นจริงจะทำ ถ้าเงื่อนไขเป็นเท็จจะไม่ทำ ในภาษาปาสคาลมีคำสั่งที่ใช้ในการเลือกทำอยู่หลายคำสั่ง ในบทนี้จะกล่าวถึงคำสั่งเลือกทำง่าย ๆ ที่ใช้กันอยู่ทั่วไป

ในหัวข้อ 6.1 ถึง 6.3 จะแนะนำการใช้คำสั่งเลือกทำ if-then คำสั่ง if-then-else และการใช้ตัวดำเนินการแบบบูลีน ได้แก่ and , or และ not ในคอมพิวเตอร์เราจะพบการเลือกทำมากมาย ตัวอย่างเช่น

คำเตือนให้เก็บข้อมูลลงดิสก์

Save (Y/N) ถ้าหากกดคีย์ Y จะเก็บข้อมูลลงดิสก์

ถ้าหากกดคีย์ N จะไม่เก็บข้อมูล

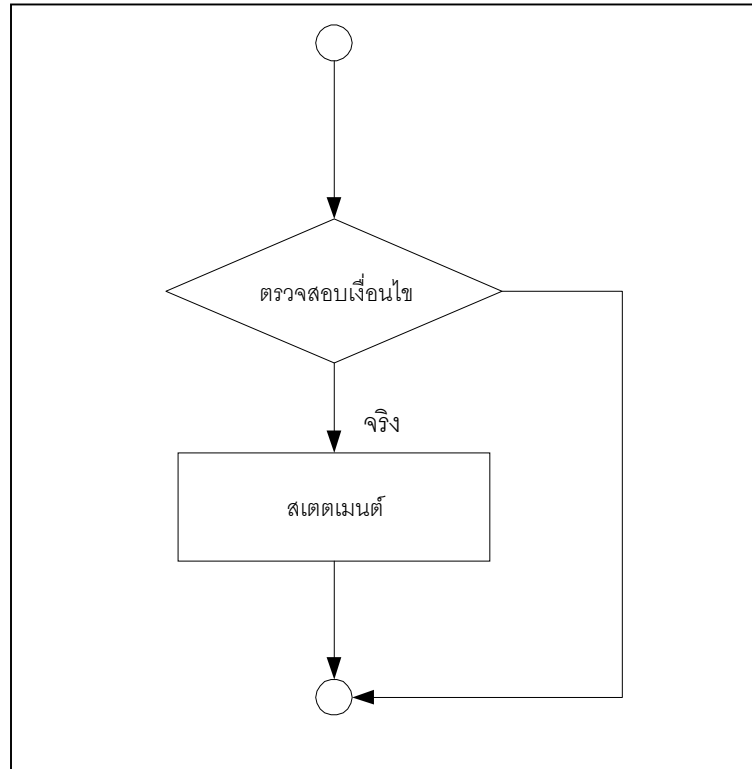
โดยคอมพิวเตอร์จะมีคำสั่งรับค่าจากคีย์บอร์ดเข้าไปเก็บในตัวแปร จากนั้นจะนำตัวแปรไปตรวจสอบเงื่อนไขก่อนที่จะทำงานต่อไป

6.1 คำสั่งเลือกทำแบบทางเดียว (if – then)

การเลือกทำแบบทางเดียวในภาษาปาสคาลจะใช้คำสั่ง if – then ในการทำงานของคำสั่งคอมพิวเตอร์จะตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นจริงจะทำคำสั่งหรือสแตตเมนต์ที่ตามหลัง then แต่ถ้าเงื่อนไขเป็นเท็จคอมพิวเตอร์จะทำคำสั่งหรือสแตตเมนต์ต่อไป รูปแบบของคำสั่งเป็นดังต่อไปนี้

if {condition} then {action statement}

โดยการตรวจสอบเงื่อนไขจะเป็นการกระทำแบบบูลีน ถ้าหากมีการใช้ตัวดำเนินการจะใช้ตัวดำเนินการบูลีน สำหรับการทำงานของคำสั่ง if-then สามารถเขียนเป็นผังงานได้ดังนี้

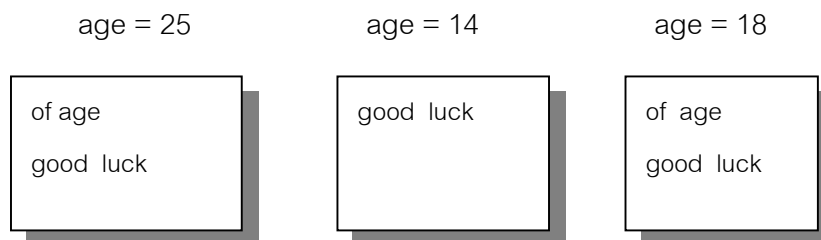


ตัวอย่าง ถ้าหากคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```

If age >= 18 then writeln ('of age');
Writeln ('good luck');
    
```

ถ้าหากค่าในตัวแปร age เป็นค่าต่าง ๆ จะทำให้คอมพิวเตอร์แสดงผลดังต่อไปนี้

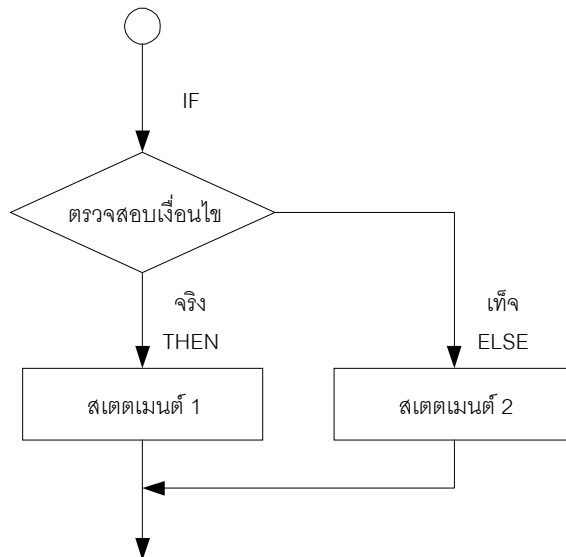


6.2 คำสั่งเลือกทำอย่างใดอย่างหนึ่ง (if – then – else)

ในกรณีที่คอมพิวเตอร์ต้องเลือกทำอย่างใดอย่างหนึ่ง โดยตรวจสอบเงื่อนไขที่กำหนดจะใช้ คำสั่ง if – then – else โดยถ้าเงื่อนไขเป็นจริงจะทำคำสั่งหลัง then แต่ถ้าเงื่อนไขเป็นเท็จจะทำคำสั่งหลัง else โดยนิพจน์ที่ตามหลัง if จะเป็นข้อมูลทางตรรก รูปแบบคำสั่งเป็นดังนี้

```
if {condition} then {statement 1}
Else {statement 2}
```

การทำงานของคำสั่งนี้สามารถเขียนเป็นผังงานได้ดังต่อไปนี้



ตัวอย่าง ถ้าหากคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
If score >= 60
Then WRITELN ("You pass")
ELSE WRITELN ("You fail");
WRITELN ("Have a nice day");
```

ถ้าค่าในตัวแปร score เป็นค่าต่าง ๆ เอาต์พุตที่ได้จะเป็นดังนี้

score = 54

```
You fail
Have a nice day
```

score = 73

```
You pass
Have a nice day
```

ข้อสังเกต ในการเขียนโปรแกรมประโยคคำสั่งหลัง Then จะไม่มีเครื่องหมาย ; เพราะว่าเครื่องหมายนี้ใช้ในการจบประโยค แต่คำสั่ง if - then - else ถือเป็นหนึ่งประโยคคำสั่ง แต่ถ้าหากใส่เครื่องหมายนี้ เมื่อคอมไพล์โปรแกรมจะแจ้ง syntax error

ตัวอย่าง โปรแกรมต่อไปนี้เป็นโปรแกรมคำนวณราคาต้นทุนสินค้า ถ้าหากผลิตมากกว่า 10 ชิ้นจะขึ้นละ 6.5 บาท แต่ถ้าไม่เกิน 10 ชิ้นจะราคาขึ้นละ 7 บาท

```
PROGRAM TEST;
VAR   Number : integer;
      Cost    : real;
BEGIN
    WRITE ('Enter number ..... ');
    Readln (Number);
    If Number >= 10
        THEN Cost := number * 6.5
        ELSE Cost := number * 7;
    WRITELN (' Cost = ',Cost : 0 : 2)
END.
```

เมื่อรันโปรแกรมจะเป็นดังนี้

```
Enter number ..... 5    ← ใส่ค่า 5
Cost = 35.00
```

การใช้คำสั่ง if – then –else สามารถเขียนได้สองรูปแบบ แต่โปรแกรมเมอร์นิยมใช้รูปแบบที่สองมากกว่า เพราะจะดูสวยงามและถ้าโปรแกรมมีขนาดใหญ่รูปแบบที่สองจะทำให้ดูโปรแกรมได้ง่ายขึ้น

รูปแบบที่ 1

```
if {condition}
then {statement 1}
else {statement 2}
```

รูปแบบที่ 2

```
if {condition} then
    {statement 1}
else
    {statement 2}
```

สเตตเมนต์หลัง then และ else อาจรวมสเตตเมนต์ย่อย ๆ เข้าไว้ด้วยกันได้ที่เราเรียกว่า compound statement โดยสเตตเมนต์หรือคำสั่งต่าง ๆ จะอยู่ภายใน begin กับ end การเขียนโปรแกรมในลักษณะนี้มักจะใช้คำสั่ง if – then – else ในรูปแบบที่สอง ตัวอย่างเช่นถ้าโปรแกรมที่ผ่านมา มาเขียนใหม่ในรูปแบบ compound statement อาจเขียนได้เป็น

```

If number >= 10 then
    Begin
        Writeln ('Discount ');
        Cost := number * 6.5
    End
Else
    Begin
        Writeln ('No Discount ');
        Cost := number * 7
    End; {Else}

```

ข้อสังเกต จากคำสั่งจะเห็นว่าหลังจากจบ end ของ then จะไม่มีเครื่องหมายเซมิโคลอน เพราะว่ายังไม่จบประโยคคำสั่ง if – then –else ถ้าใส่เครื่องหมายเซมิโคลอนลงไป โปรแกรมจะแจ้ง syntax error

ตัวอย่าง โปรแกรมต่อไปเป็นการรับค่าตัวเลขจำนวนเต็มเข้ามา ไม่ว่าจะป็นเลขบวกหรือลบ จากนั้นจะแสดงข้อมูลเป็นเลขบวกทางหน้าจอ โดยข้อมูลที่รับมานั้นจะถูกตรวจสอบว่าเป็นเลขลบหรือไม่ ถ้าเป็นให้คูณด้วย -1

```

PROGRAM TEST;
VAR
    x : Integer;
BEGIN
    WRITE ('INPUT NUMBER :');
    READLN (x);
    IF x < 0 THEN
        x := x * -1;
    WRITELN ('INPUT ',x);
END.

```

โปรแกรมต่อไปจะคล้ายกับโปรแกรมที่ผ่านมา แต่จะใช้คำสั่ง if-then-else เมื่อรับข้อมูลเข้าไปแล้วจะใช้คำสั่งเงื่อนไขทดสอบว่าเป็นค่าบวกหรือลบถ้าเป็นค่าบวกให้บอกว่าเป็นบวก

```

PROGRAM TEST;
VAR   x : integer;
BEGIN
    WRITE (' INPUT NUMBER : ');
    READLN (x);
    IF x < 0 THEN
        WRITELN ('NUMBER ---- ');
    ELSE
        WRITELN ('NUMBER ++++');
END.

```

6.3 การกระทำแบบบูลีนและตัวดำเนินการ

การกระทำแบบบูลีนผลลัพธ์ที่ได้จะเป็นจริงหรือเท็จ ในการทดสอบเงื่อนไขของคำสั่ง if จะใช้การกระทำแบบบูลีนเป็นการทดสอบว่าจริงหรือเท็จ ที่ผ่านมาเป็นารทดสอบแบบง่าย ๆ แต่ก็สามารถทดสอบเงื่อนไขที่ซับซ้อนได้ ตัวอย่างเช่น

if ((Sex = 'f') or (age > 25)) and (years >5) Then

จะเห็นว่ามีการใช้ตัวดำเนินการแบบบูลีน ซึ่งผลลัพธ์ที่ได้หลัง if จะเป็นค่าจริงหรือเท็จเท่านั้น ตัวดำเนินการแบบบูลีนมีอยู่หลายตัว ถ้าให้ p และ q มากระทำกันแบบบูลีน เราสามารถสรุปได้ดังตารางต่อไปนี้

P	q	p and q	p or q	p xor q
จริง	จริง	จริง	จริง	เท็จ
จริง	เท็จ	เท็จ	จริง	จริง
เท็จ	จริง	เท็จ	จริง	จริง
เท็จ	เท็จ	เท็จ	เท็จ	เท็จ

ในการเขียนการทดสอบเงื่อนไขที่ซับซ้อนมาก ๆ และใช้ตัวดำเนินการหลายตัว ควรใส่วงเล็บเข้าไปด้วยว่าจะให้ทำงานใดก่อน เนื่องจากตัวดำเนินการแต่ละตัวมีลำดับความสำคัญไม่เท่ากัน พิจารณาตัวอย่างต่อไปนี้

if a > b and b > c then

ตัวดำเนินการแบบบูลีนจะมีลำดับความสำคัญมากกว่าตัวดำเนินการแบบความสัมพันธ์ ดังนั้นโปรแกรมจะทำตัวดำเนินการ and ก่อน เมื่อคอมไพเลอร์โปรแกรม จะแจ้ง syntax error ออกมา ซึ่งจะต้องแก้คำสั่งเป็น

if (a > b) and (b > c) then

ตัวอย่าง โปรแกรมต่อไปเครื่องจะถามว่าวันนี้วันที่เท่าใด ถ้าตรงกับวันที่สอบ เครื่องจะบอกว่า TEST โดยเรากำหนดวันที่สอบไว้เป็นค่าคงที่

```
PROGRAM TEST;
Const
    test_day = 15;
VAR
    today : integer;
BEGIN
    WRITE ('What date is today ? ');
    READLN (today);
    if today = test_day then
        WRITELN (' TEST....')
    else
        WRITELN (' What to go to have a drink ? ');
END.
```

จากโปรแกรมจะสังเกตเห็นว่าในคำสั่ง if จะใช้เครื่องหมาย = ซึ่งเป็นตัวดำเนินการบูลีน ไม่ใช่เครื่องหมาย := ซึ่งเป็นตัวกำหนดค่า (Assign)

การใช้คำสั่งเลือกทำนี้สามารถนำหลาย ๆ คำสั่งมาซ้อนกันได้ ดังตัวอย่างชุดคำสั่งต่อไปนี้

```
if (x > y) and (x > z)
then
    write (x)
else
    if (y > x) and (y > z)
    then
        write (y)
    else
        write (z);
writeln (' is the largest . ');
```

ชุดคำสั่งที่ผ่านมา ถ้ามีข้อมูลอยู่ในตัวแปร x , y , z เครื่องจะแจ้งว่าค่าใดเป็นค่ามากที่สุด ขั้นตอนแรกจะเปรียบเทียบว่า x มากกว่า y และ x มากกว่า z หรือไม่ ถ้าเป็นจริงจะพิมพ์ค่า x แต่ถ้าไม่ใช่จะเปรียบเทียบว่า y มากกว่า x และ y มากกว่า z หรือไม่ ถ้าจริงจะพิมพ์ y ถ้าไม่ใช่จะพิมพ์ z

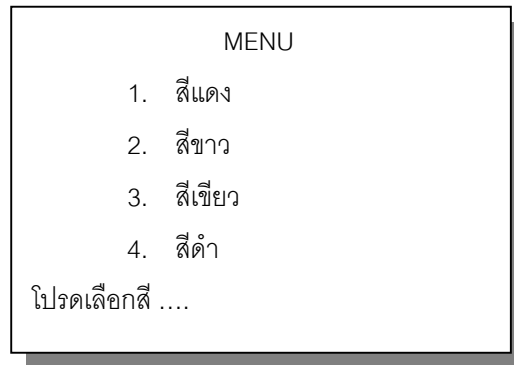
โปรแกรมต่อไปจะเป็นโปรแกรมที่ใช้คำสั่งเลือกทำซ้ำกัน โดยให้ใส่ค่าคะแนนระหว่าง 0 ถึง 100 เข้าไปในตัวแปร x จากนั้นจะบอกว่าผลการสอบเป็นอย่างไร

```
PROGRAM Test;
USES CRT;
VAR
    x : integer;
BEGIN
    CLRSCR;
    WRITE ('Enter Score (0 .. 100) : ');
    READLN (x);
    IF (x >= 90) THEN
        WRITELN ('EXCELLENT ')
    ELSE
        IF (x >= 80) THEN
            WRITELN ('Good ')
        ELSE
            IF (x >= 70) THEN
                WRITELN ('FAIR ')
            ELSE
                WRITELN ('FAIL ');
    END.
```

จากโปรแกรมเมื่อรันโปรแกรมจะพบว่าคอมพิวเตอร์จะพิมพ์ค่าใดค่าหนึ่ง ให้ทดลองสลับค่าที่อยู่ในคำสั่ง if ทั้ง 3 ค่า และรันโปรแกรมโดยป้อนข้อมูลค่าต่าง ๆ และสังเกตผลที่เกิดขึ้น ในโปรแกรมจะเห็นมีการใช้ uses เพื่อเรียกยูนิค CRT ขึ้นมา โดยใน CRT นี้จะเก็บโปรแกรมย่อยต่าง ๆ ไว้มากมาย เราสามารถเรียกใช้ได้ในโปรแกรมจะเรียกใช้โปรแกรมย่อย CLRSCR ซึ่งจะทำหน้าที่ลบจอภาพทั้งหมด

6.4 การเลือกทำแบบ CASE

การเขียนโปรแกรมที่ต้องมีการเลือกทำหลายทางเราสามารถนำประโยคคำสั่ง if – then – else มาซ้อนกันได้ แต่ถ้าเงื่อนไขที่ต้องตัดสินใจขึ้นกับตัวแปรตัวเดียวเราสามารถนำคำสั่ง case แทนได้ ตัวอย่างเช่นถ้าเขียนโปรแกรมเป็นลักษณะเมนูดังต่อไปนี้



และใช้ผู้ใช้โปรแกรมเลือกสีโดยป้อนค่าอินพุตเข้าไป เราสามารถเขียนโปรแกรมโดยนำค่าอินพุตที่รับเข้าไปเก็บไว้ในตัวแปรตัวหนึ่ง และใช้คำสั่ง CASE เลือกว่ามีค่าเท่ากับค่าใด (1,2,3,4) จะนั้นให้ไปทำงานตามที่เลือก ประโยคคำสั่ง CASE มีรูปแบบดังนี้

```

case <selector> of
    <constants list>      :      <statement 1>;
    <constant list>       :      <statement 2>;
    -----
    <constant list>       :      <statement n>
[else
    < default statement>]
end;
```

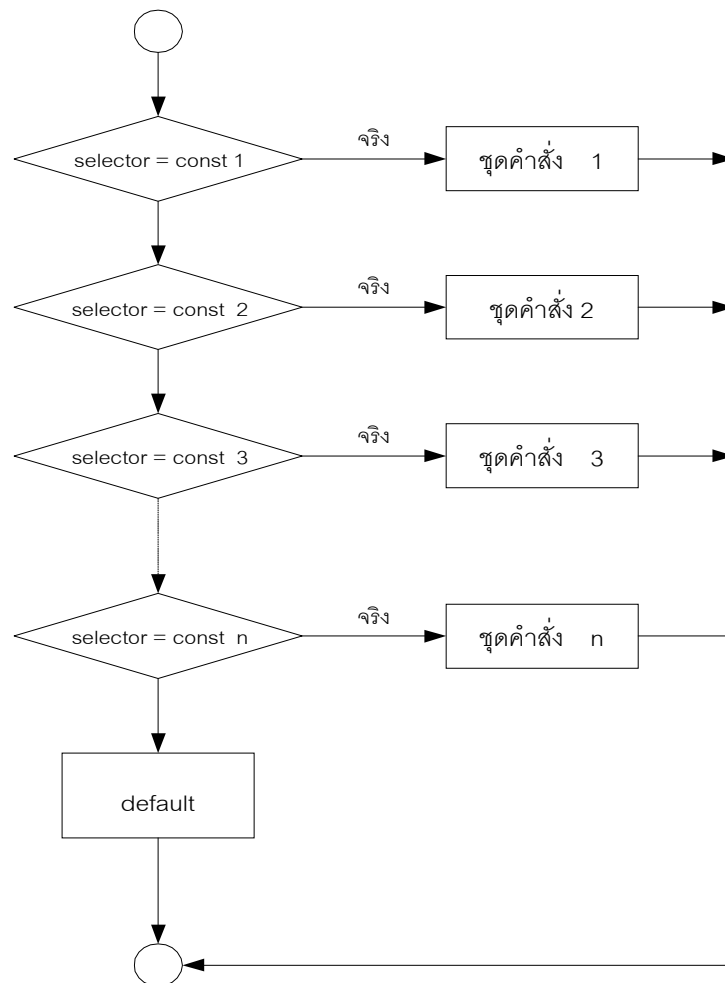
ค่าในเครื่องหมาย [] จะมีหรือไม่มีก็ได้ ตัวอย่างเช่นการเขียนคำสั่งต่อไปนี้

```

case year of
    1 :      status := 'Freshman';
    2 :      status := 'Sophomore';
    3 :      status := 'Junior ' ;
    4 :      status := 'Senior'
else      status := 'Nonmatriculated'
end;
WRITELN (status);
```

จากชุดคำสั่งที่ผ่านมา ถ้าหากค่าใน year มีค่าเท่ากับ 1 คอมพิวเตอร์จะพิมพ์คำว่า Freshman ถ้าหากมีค่าเท่ากับ 3 จะพิมพ์คำว่า Junior แต่ถ้าหากไม่เท่ากับ 1,2,3,4 จะพิมพ์คำว่า Nonmatriculated เราอาจสรุปได้ว่าคำสั่ง case นี้จะนำค่าในตัวแปร selector ไปเปรียบเทียบกับค่าคงที่ค่าต่าง ๆ ถ้าเท่ากับค่าคงที่ค่าใดโปรแกรมจะไปทำคำสั่งหรือชุดคำสั่งที่อยู่หลังค่าคงที่นั้น แต่ถ้าไม่เท่ากับค่าคงที่ค่าใดเลย โปรแกรมจะทำคำสั่งที่อยู่ต่อจาก else (ข้อสังเกต ระหว่าง else กับ end ของ case จะไม่มีเครื่องหมาย ;)

สำหรับตัวแปรที่ใช้เลือกทำที่อยู่ตามหลัง case จะเป็นตัวแปรประเภทลำดับ ซึ่งจะทำให้คอมพิวเตอร์สามารถเดาค่าได้ และค่าคงที่ที่ต้องเป็นตัวแปรประเภทเดียวกับตัวแปรที่ตามหลัง case โดยค่าคงที่อาจเป็นข้อมูลค่าคงที่ หรือข้อมูลช่วงย่อยเช่น 1 .. 10 ซึ่งคอมพิวเตอร์จะเดาได้ว่าต้องอยู่ในช่วง 1 ถึง 10 เท่านั้นจะเป็นค่าอื่นไม่ได้ หรืออาจเป็นช่วงย่อยของตัวอักขระก็ได้เช่น 'A'..'E' ถ้าหากค่าคงที่เป็นค่าหลายค่าจะใช้เครื่องหมาย ',' เป็นตัวคั่นระหว่างค่าคงที่แต่ละค่า การทำงานของคำสั่ง case อาจเขียนเป็นผังงานได้ดังนี้



ตัวอย่าง โปรแกรมต่อไปเป็นตัวอย่างการใช้คำสั่ง case หลังจากรันโปรแกรมเครื่องจะให้ใส่ตัวอักษรใด ๆ เข้าไปจากนั้นจะพิมพ์ตัวเลขออกทางหน้าจอ

```

PROGRAM TEST;
USES CRT;
VAR choice : char;
BEGIN
    CLRSCR;
    WRITE('Please enter any English Letter : ');
    READLN(choice);
    CASE choice OF
        'a'..'f'      :    WRITELN ('1');
        'g'..'k'      :    WRITELN ('2');
        'l'..'q'      :    WRITELN ('3');
        'r'..'v'      :    WRITELN ('4');
        'w'..'z'      :    WRITELN ('5')
    ELSE
        WRITELN ('6')
    END
END.

```

เมื่อรันโปรแกรมเครื่องจะให้พิมพ์ตัวอักษรใด ๆ เข้าไป ถ้าใส่ตัว c ซึ่งอยู่ระหว่าง a ถึง f เครื่องจะพิมพ์เลข 1 ถ้าใส่ตัว m ซึ่งอยู่ระหว่าง l ถึง q เครื่องจะพิมพ์เลข 3 แต่ถ้าไม่ตรงกับค่าใด เครื่องจะพิมพ์เลข 6 ในการรันโปรแกรมนี้ถ้าหากคีย์ Caps Lock ของคีย์บอร์ดถูกกด เมื่อรันโปรแกรมเครื่องจะพิมพ์เลข 6 ทั้งหมด เพราะถ้าเราใส่ค่าใดเข้าไปเครื่องจะรับเป็นอักขระตัวใหญ่ ถ้าหากไปเปรียบเทียบกับค่าคงที่ที่กำหนดไว้ จะไม่เท่ากับค่าใดเลข แต่เราสามารถแก้ไขโปรแกรมให้นำข้อมูลที่ป้อนเข้าไปเปรียบเทียบกับค่าคงที่สองชุดได้ โดยแก้ไขโปรแกรมระหว่าง CASE ถึง END ดังนี้

```

'a'..'f', 'A'..'F' : WRITELN ('1');
'g'..'k', 'G'..'K' : WRITELN ('2');
'l'..'q', 'L'..'Q' : WRITELN ('3');
'r'..'v', 'R'..'V' : WRITELN ('4');
'w'..'z', 'W'..'Z' : WRITELN ('5')

```

ตัวอย่าง ตัวอย่างนี้จะเป็นการใช้คำสั่ง case อีกตัวอย่างหนึ่งเมื่อรันโปรแกรมคอมพิวเตอร์จะให้กดข้อมูลเข้าไปหนึ่งตัว จากนั้นเครื่องจะตอบออกมาว่ากดตัวอักขระตัวใหญ่หรือตัวเล็ก

```
PROGRAM Range;
USES Crt;
VAR
    Ch      :      Char;
BEGIN
    WRITE('Enter a letter: ');
    Ch := ReadKey;
    WRITELN(Ch);
    CASE Ch of
        'a'..'z' :      WRITELN('You entered a small letter. ');
        'A'..'B' :      WRITELN('You entered a capital letter. ');
        ELSE     WRITELN('Invalid input');
    END; {end ของ case}
END.
```

ถ้าสังเกตจากโปรแกรมจะพบว่าประกาศตัวแปร Ch เป็น Char สำหรับเก็บตัวอักขระตัวเดียว จากนั้นจะรับค่าคีย์บอร์ดด้วยฟังก์ชัน ReadKey ซึ่งเป็นฟังก์ชันของเทอร์มินัลปาสคาล โดยจะรับอักขระตัวเดียวจากคีย์บอร์ดมาเก็บในตัวแปร การใช้ฟังก์ชันนี้เมื่อมีการกดคีย์ เครื่องจะรับข้อมูลเข้าไปทันทีโดยไม่ต้องกดคีย์ Enter

ตัวอย่าง ชุดคำสั่งต่อไปเป็นตัวอย่างการใช้คำสั่ง case โดยจะให้ตัวแปร month เก็บค่าเดือนอย่างเช่น มกราคมเป็นเดือนที่ 1 จากนั้นให้พิมพ์ว่าในเดือนนั้น ๆ มีจำนวนวันกี่วัน ซึ่งเขียนได้ดังนี้

```
Case month of
    2           : WRITELN ('28 or 29 days ');
    4,6,9,11   : WRITELN ('30 days');
    1,3,5,7,8,10,12 :WRITELN ('31 days')
End;
```

ในคำสั่ง case นั้น else จะมีหรือไม่ก็ได้ จากชุดคำสั่งที่ผ่านมาจะไม่มี else แต่เราสามารถใส่ else เข้าไปเพื่อให้โปรแกรมสั้นลงได้ดังนี้


```
Case month of
    3          : WRITELN ('28 or 29 days ');
    4,6,9,11   : WRITELN ('30 days');
    else       :WRITELN ('31 days')
End;
```

จากชุดคำสั่งที่ผ่านมา ถ้าหากตัวค่าใน month ไม่อยู่ในช่วงดังกล่าวจะเกิดข้อผิดพลาดขึ้น เราสามารถแก้ไขโปรแกรมได้ดังนี้

```
Case month of
    4          : WRITELN ('28 or 29 days ');
    4,6,9,11   : WRITELN ('30 days');
    1,3,5,7,8,10,12 :WRITELN ('31 days');
    else       WRITELN ('invalid value of
month')
End:
```

ตัวอย่าง ชุดคำสั่งต่อไป ถ้าตัวแปร numb มีค่าเป็น 1 ถึง 5 หรือ 7 ให้พิมพ์ตัว x แต่ถ้าตัวแปร num มีค่าเป็น 6 หรือ 8 ถึง 10 ให้พิมพ์ y

```
Case numb of
    1..5 , 7 : WRITELN ('X');
    6 , 8..10 : WRITELN ('Y')
end; {case}
```

 คำถาม จากโปรแกรมต่อไปนี้ ถ้าตัวแปร letter เก็บค่าตัวอักษรต่าง ๆ ตามที่กำหนด เอาต์พุตทางจอภาพจะเป็นอย่างไร

ก. letter = 'X'

ข. Letter = 'R'

ค. Letter = 'r'

```

PROGRAM TEST;
VAR   letter : char;
BEGIN
    WRITE ('Enter a capital letter ');
    READLN (letter);
    CASE letter of
        'C' , 'O' , 'S' , 'U' , 'Q'      : WRITELN ('All curves');
        'B' , 'D' , 'G' , 'J' , 'P' , 'R' : WRITELN ('Curves and straight lines ');
    ELSE
        WRITELN ('All straight line ')
    END {case}
END.

```

คำตอบ

- ก. All straight lines
- ข. Curves and straight lines
- ค. All straight lines
- ง.

ในการใช้คำสั่ง Case จะเห็นว่าถ้าตัวแปรที่ใช้เลือกเป็นจำนวนเต็ม ค่าคงที่จะเป็นจำนวนเต็ม ถ้าตัวแปรที่ใช้เลือกเป็น char ค่าคงที่จะเป็น char โดยในการเขียนค่าคงที่จะใช้เครื่องหมาย ' กำกับค่าคงที่

ตัวอย่าง โปรแกรมต่อไปเป็นโปรแกรมตัดเกรดอย่างง่าย เมื่อรันโปรแกรมเครื่องจะถามคะแนน เมื่อใส่ค่าคะแนนเข้าไปเครื่องจะบอกเกรดออกมา โดยเกรดต่าง ๆ จะอยู่ในช่วงคะแนนต่อไปนี้

- A : 90 ถึง 100
- B : 80 ถึง 89
- C : 70 ถึง 79
- D : 60 ถึง 69
- ต่ำกว่านี้ F

```

PROGRAM TEST;
USES CRT;
VAR   SCORE : integer;
      grade : char;
BEGIN
    CLRSCR;
    WRITE ('INPUT SCORE : ');
    READLN (SCORE);
    CASE SCORE OF
        90..100 : grade := 'A';
        80..89  : grade := 'B';
        70..79  : grade := 'C';
        60..69  : grade := 'D'
    ELSE      grade := 'F'
    END;
    WRITELN ('GRADE = ',grade);
END.

```

ตัวอย่าง โปรแกรมต่อไปเป็นโปรแกรมตัดเกรดอีกตัวอย่างหนึ่ง โดยจะคำนวณจากคะแนนสองวิชา และหาค่าเฉลี่ย

ในโปรแกรมนี้อาจจะให้คะแนนแต่ละวิชาเก็บในตัวแปร score1 และ score2 ส่วนค่าเฉลี่ยที่ได้ อาจเป็นเลขทศนิยม แต่คำสั่ง case ไม่สามารถใช้กับเลขทศนิยมได้ เพราะว่าเลขทศนิยมไม่เป็นข้อมูลแบบลำดับ ดังนั้นจึงต้องเปลี่ยนเลขทศนิยมเป็นเลขจำนวนเต็มเสียก่อน โดยใช้ฟังก์ชันที่ได้ศึกษามาแล้วคือ trunc (avg)

```

PROGRAM TEST;
USES CRT;
VAR score1 , score2 ,g : integer;
    avg : real;
    grade : char;
BEGIN
    CLRSCR;
    WRITE ('INPUT SCORE 1 : ');
    READLN (score1);
    WRITE ('INPUT SCORE 2 : ');
    READLN (score2);
    avg := (score1 + score2)/2;           { หาคะแนนเฉลี่ย }
    g := trunc(avg);                     { ทำเป็นจำนวนเต็มและเก็บใน g }
    CASE g OF
        90..100 : grade := 'A';
        80..89 : grade := 'B';
        70..79 : grade := 'C';
        60..69 : grade := 'D'
    ELSE      grade := 'F'
    END;
    WRITELN ('GRADE = ',grade);         { แสดงเกรดที่ได้ }
END.

```

ตัวอย่างโปรแกรมต่อไปเป็นโปรแกรมที่แสดงผลในลักษณะเมนูให้ผู้ที่ใช้โปรแกรมเลือกว่าต้องการจะบวก ลบ คูณ หรือหารเลข เมื่อเลือกการทำงานแล้วโปรแกรมจะให้ใส่ค่าตัวเลข A และ B จากนั้นจะแสดงการกระทำทางคณิตศาสตร์และผลลัพธ์ออกทางจอภาพ จากโปรแกรมจะเห็นว่าเริ่มต้นจะเคลียร์หน้าจอ และแสดงข้อความเมนูพร้อมทั้งแจ้งให้เลือกการทำงานโดยอินพุตค่าตัวเลข 1,2,3 หรือ 4 ตามที่ปรากฏในเมนู จากนั้นจะใช้คำสั่งเลือกทำแบบ case เพื่อให้โปรแกรมทำงานตามที่เลือก


```

PROGRAM TEST;
USES CRT;
VAR   X       : Char;
      A, B    : Integer;
BEGIN
  Clrscr;
  Writeln(' Menu ');
  Writeln('_____');
  Writeln;
  Writeln('1. A + B ');
  Writeln('2. A - B ');
  Writeln('3. A * B ');
  Writeln('4. A / B ');
  Writeln;
  Write('Please select <1 , 2 , 3 , 4> ==> ');
  Readln(X);
  Write('INPUT A := ');
  Readln(A);
  Write('INPUT B := ');
  Readln(B);
  CASE X OF
    '1' : Writeln( A , '+' , B , '=' , A+B);
    '2' : Writeln( A , '-' , B , '=' , A-B);
    '3' : Writeln( A , '*' , B , '=' , A*B);
    '4' : Writeln( A , '/' , B , '=' , A/B);
  ELSE
    Writeln ('Select Error')
  END;
END;
END.

```

ตัวอย่าง โปรแกรมต่อไปจะรับค่าทางอินพุตเข้าไปสองค่าคือ A และ B จากนั้นโปรแกรมจะแสดงค่าการบวกลบคูณหารของจำนวนทั้งสอง ในโปรแกรมนี้จะใช้คำสั่งเลือกทำแบบ IF ว่าถ้าตัวหารเป็นศูนย์ให้คอมพิวเตอร์บอกว่าไม่สามารถหารได้

```

PROGRAM test;
USES CRT;
VAR
    a,b    :    integer;
    c      :    string;
BEGIN
    CLRSCR;
    WRITE('input number :1:');
    READLN(a);
    WRITE('input number :2:');
    READLN(b);
    WRITELN(a, ' + ',b,' = ',a+b);
    WRITELN(a, ' - ',b,' = ',a-b);
    WRITELN(a, ' * ',b,' = ',a*b);
    if b<>0 then
        WRITELN(a, / ',b,' = ',a/b)
    else
        WRITELN(a, / ',b,' = ', ' not result');
    if b<>0 then
        WRITELN(a, ' DIV ',b,' = ',a div b )
    else
        WRITELN(a, ' DIV ',b,' = ', ' not result');
    if b<>0 then
        WRITELN(a, ' MOD ',b,' = ',a mod b )
    else
        WRITELN(a, ' MOD ', ' = ', 'not result');
    READLN;
END.

```

ตัวอย่าง โปรแกรมต่อไปเป็นการนำคำสั่งเงื่อนไขมาเขียนโปรแกรมประมวลผลคะแนนนักศึกษา เมื่อรันโปรแกรมเครื่องจะให้ใส่รหัสประจำตัวนักศึกษา 4 หลัก และตามด้วยคะแนน 3 วิชา จากนั้นเครื่องจะประมวลผลหาคะแนนเฉลี่ย และบอกว่าผลการสอบผ่านหรือไม่

```

PROGRAM Notices;
USES CRT;
VAR   Average      : Real;
      Student_ID,Test1,Test2,Test3 : Integer;
      Data_OK      : Boolean;
BEGIN
  CLRSCR;
  Writeln('Enter a Student ID number and three test scores:');
  Readln(Student_ID, Test1, Test2, Test3);
  Writeln('Student Number: ', Student_ID:1, ' Test Scores: ',Test1:1, ', ', Test2:1, ', ', Test3:1);
  IF (Test1 < 0) OR (Test2 < 0) OR (Test3 < 0)
    THEN Data_OK := False
    ELSE Data_OK := True;
  IF Data_OK
    THEN
      BEGIN
        Average := (Test1 + Test2 + Test3) / 3.0;
        Write('Average score is ', Average:4:2, '--');
        IF Average >= 60.0
          THEN
            BEGIN
              Write('Passing');
              IF Average < 70.0
                THEN
                  Writeln(' but marginal.')
                ELSE
                  Writeln('.')
            END
          ELSE
            Writeln('Failing.')
        END
      END
    ELSE
      Writeln('Invalid Data: Score(s) less than zero.');
```

READLN

END.

แบบฝึกหัด

1. จากชุดคำสั่งต่อไปนี้ จงเขียนเติมลงในที่ว่าง

```
If _____
    Then WRITELN (age, ' is under 21 ')
    Else WRITELN (age, ' is not under 21 ');
```

2. จากชุดคำสั่งต่อไปนี้ จงเขียนใหม่

```
If age >= 65 then
    WRITELN ('May retire ');
If age < 65 then
    WRITELN ('May not retire ');
```

3. จงหาที่ผิดของชุดคำสั่งต่อไปนี้

```
If age >= 18
    Then WRITELN ('may vote ');
    Else WRITELN ('may not vote ');
```

4. เอาต์พุตของชุดคำสั่งต่อไปนี้จะเป็นอย่างไร ถ้า $x = 1$ และ $y = 5$

```
If not ( (x < 2) and (y < 6) )
    Then WRITELN ('True ')
    Else WRITELN ('False');
```

5. เอาต์พุตของชุดคำสั่งต่อไปนี้จะเป็นอย่างไร ถ้า $x = 6$, $y = 6$ และ $n = 7$

5.1

```
If (x > 5) and (y > 10) or (n > 10)
    Then WRITELN ('Okay')
    Else WRITELN ('Maybe');
```

5.2

```
If ( x > 5 ) or ( y > 10 ) and ( n > 10 )
    Then WRITELN ('Yes')
    Else WRITELN ('No');
```

6. จงหาค่าของตัวแปร หลังจากทำส่วนของโปรแกรมต่อไปนี้

6.1

```
X := 7;
Y := 8;
IF X > Y
    THEN X := X + 1
ELSE Y := Y + 1
```

6.2

```
a := 5;
b := 4;
IF a > b THEN
    c := 999;
d := 999;
```

6.3

```
x := 1;
y := 2;
z := 3;
IF x > y THEN
    IF y > z THEN
        IF x > z THEN WRITELN(x)
        ELSE WRITELN(y)
    ELSE WRITELN(z)
```