

บทที่ 3

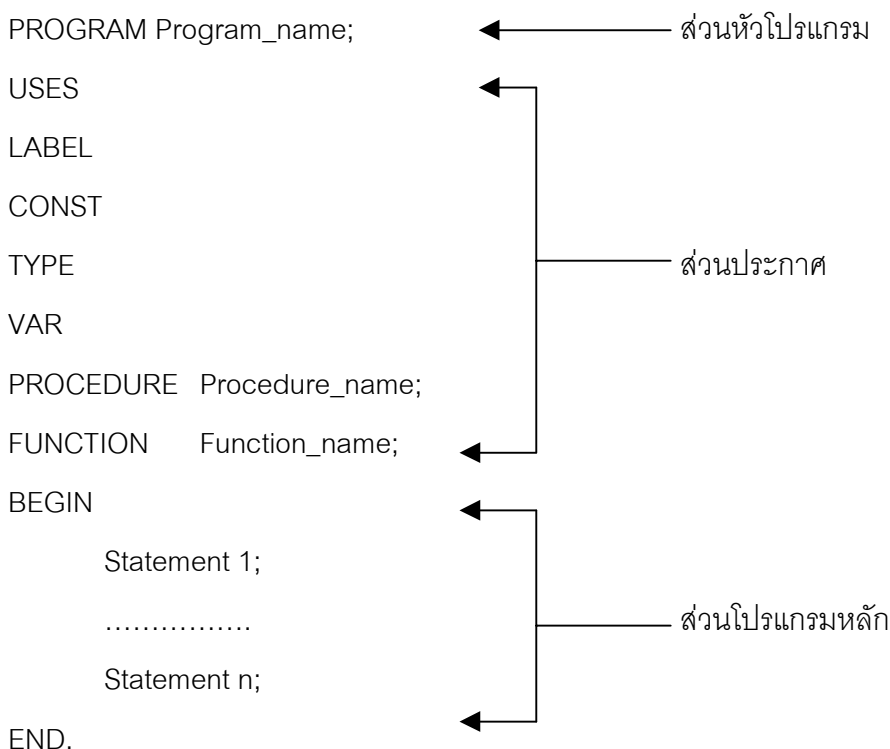
โครงสร้างภาษาปาสคาลเบื้องต้น Introduction to Pascal Structure

3.1 โครงสร้างโปรแกรม

การเขียนโปรแกรมแต่ละภาษานั้นโครงสร้างของโปรแกรมจะต่างกัน ในบทนี้จะกล่าวถึงโครงสร้างของภาษาปาสคาล รวมทั้งการเขียนโปรแกรมอย่างง่าย ลักษณะโครงสร้างของภาษาปาสคาลแบ่งออกได้เป็น 3 ส่วนดังต่อไปนี้

1. ส่วนหัวโปรแกรม (Program Header Part)
2. ส่วนประกาศ (Declaration Part)
3. ส่วนโปรแกรมหลัก (Program Body Part)

โครงสร้างทั้งสามส่วนจะมีรูปแบบดังต่อไปนี้



จะเห็นว่าโครงสร้างของโปรแกรมประกอบด้วยหลายส่วน แต่ในการเขียนโปรแกรมนั้นไม่จำเป็นต้องเขียนหมดทุกส่วน ส่วนใดไม่ใช้ก็สามารถตัดทิ้งได้ แต่ทุกโปรแกรมต้องมีส่วนหัวโปรแกรม (PROGRAM ตามด้วยชื่อ) และส่วนโปรแกรมหลัก (BEGIN กับ END)

โปรแกรมแบบง่ายที่สุดอาจเขียนได้ดังนี้

```
PROGRAM TEST;                               ;ส่วนหัวโปรแกรมบอกว่าโปรแกรมชื่อ TEST
BEGIN
    WRITELN('COMPUTER');                     ;ส่วนโปรแกรมหลักให้พิมพ์คำว่า COMPUTER
END.
```

เมื่อรันโปรแกรมบนหน้าจอคอมพิวเตอร์จะแสดงคำว่า COMPUTER

3.1.1 ส่วนหัวโปรแกรม (Program Header Part)

ส่วนนี้ทุกโปรแกรมต้องมี จะใช้สำหรับกำหนดชื่อโปรแกรม โดยจะต้องเริ่มต้นด้วยคำว่า PROGRAM และตามด้วยชื่อโปรแกรมและปิดท้ายด้วยเครื่องหมายเซมิโคลอน (;) ตัวอย่างเช่น

```
PROGRAM Test;
```

เป็นการบอกว่าโปรแกรมนี้ชื่อ Test ส่วนเครื่องหมาย ; ใช้สำหรับบอกว่าจบคำสั่ง

3.1.2 ส่วนประกาศ (Declaration Part)

ส่วนนี้จะอยู่ต่อจากส่วนหัวโปรแกรม ส่วนนี้บางโปรแกรมอาจไม่มีก็ได้ หน้าที่ของส่วนนี้จะใช้กำหนดประเภทของข้อมูลโดยใช้ Type ใช้กำหนดตัวแปรโดยใช้ VAR ใช้กำหนดค่าคงที่โดยใช้ CONST ใช้กำหนดโปรแกรมย่อยหรือโปรซีเยอร์(Procedure) และใช้กำหนดฟังก์ชัน สำหรับรายละเอียดต่าง ๆ จะได้กล่าวต่อไป

3.1.3 ส่วนโปรแกรมหลัก (Program Body Part)

ส่วนนี้ทุกโปรแกรมจะต้องมี จะประกอบไปด้วยประโยคคำสั่งต่าง ๆ ที่จะให้โปรแกรมทำงาน โดยนำคำสั่งต่าง ๆ มาต่อเรียงกัน แต่ละประโยคคำสั่งจะจบด้วยเครื่องหมายเซมิโคลอน (Semi colon ;) โดยโปรแกรมหลักนี้จะเริ่มต้นด้วย BEGIN และจบด้วย END ตามด้วยเครื่องหมายจุด '.'

3.2 โปรแกรมเบื้องต้น

ดังที่กล่าวมาแล้วว่าส่วนประกอบของโปรแกรมมีหลายส่วน ส่วนใดไม่ใช่สามารถตัดทิ้งได้ แต่ทุกโปรแกรมจะต้องมีส่วนหัวโปรแกรมและโปรแกรมหลัก พิจารณาโปรแกรมต่อไปนี้

```
PROGRAM EX1;
BEGIN
    WRITRLN('I LOVE YOU ');
    WRITELN('COMPUTER ');
END.
```

จะเห็นว่าโปรแกรมนี้มีชื่อว่า EX1 ส่วนในโปรแกรมหลักจะประกอบด้วยคำสั่งสองคำสั่ง แต่แต่ละคำสั่งจะจบด้วยเครื่องหมาย ; คำสั่ง WRITELN นั้นจะให้คอมพิวเตอร์พิมพ์ข้อมูลที่อยู่ในวงเล็บออกทางจอภาพ เมื่อให้คอมพิวเตอร์รันโปรแกรมจะปรากฏคำว่า I LOVE YOU และ COMPUTER บนหน้าจอ

พิจารณาโปรแกรมต่อไปนี้

```
PROGRAM TEST1;
VAR    feet ,inches : integer;

BEGIN
    feet := 6;
    inches := feet * 12;
    WRITELN('Height in inches is ',inches)
END.
```

เมื่อรันโปรแกรมนี้บนจอภาพคอมพิวเตอร์จะปรากฏคำว่า

Height in inches is 72

บรรทัดแรกของโปรแกรมจะบอกชื่อโปรแกรม ส่วนบรรทัดที่สองจะเป็นการประกาศตัวแปร ขึ้นมาสองตัว โดยใช้ VAR ในการประกาศตัวแปร โดยกำหนดให้ตัวแปรชื่อ feet และ inches เป็นตัวแปรประเภท Integer ในส่วนของโปรแกรมหลักจะเริ่มต้นด้วย BEGIN และสแตตเมนต์ถัดไปจะเป็นการกำหนดค่าให้ feet ให้มีค่าเป็น 6 ส่วนสแตตเมนต์ต่อมาจะนำค่า feet คูณด้วย 12 และเก็บค่าที่ได้ในตัวแปร inches ส่วนสแตตเมนต์ WRITELN จะใช้สำหรับพิมพ์ค่าเอาต์พุตทางจอภาพ

ในส่วนของ การประกาศตัวแปรชื่อของตัวแปรจะต้องเป็นไปตามกฎการตั้งชื่อซึ่งจะได้กล่าวต่อไป ถ้าหากชื่อตัวแปรมีความยาวมากกว่า 63 ตัวอักษร โปรแกรมจะรับรู้เพียง 63 ตัวแรกเท่านั้น และในการประกาศตัวแปรถ้าหากมีตัวแปรมากกว่า 1 ตัวจะใช้เครื่องหมาย , ขึ้น การกำหนดตัวแปรจะต้องใช้คำว่า VAR ในโปรแกรมจะประกาศตัวแปรดังนี้

```
VAR feet , inches : integer
```

เป็นการบอกว่ามีตัวแปรสองตัว และเป็นตัวแปรประเภท Integer ดังนั้นตัวแปรทั้งสองตัวนี้จะใส่เลขจำนวนเต็มใด ๆ ก็ได้ แต่ต้องอยู่ในช่วง -32,768 ถึง 32,767

ในการกำหนดค่าให้กับตัวแปรจะใช้เครื่องหมาย := ตามด้วยค่าที่จะให้กับตัวแปร รูปแบบทั่วไปของการกำหนดค่าให้กับตัวแปรจะเป็นดังนี้

```
variable := expression;
```

โดยที่ expression อาจเป็นค่าคงที่ ตัวแปร หรือการกระทำทางคณิตศาสตร์ก็ได้ การทำงานของโปรแกรมจะทำค่าที่อยู่ทางขวาของเครื่องหมาย assignment (:=) มาใส่ให้กับตัวแปรที่อยู่ทางซ้าย ในภาษา Pascal สามารถใช้เครื่องหมายการกระทำทางคณิตศาสตร์บวก(+) และ (-) ได้ ส่วนการคูณจะใช้เครื่องหมาย * และการหารจะใช้เครื่องหมายสองตัวคือ div และ / ส่วนรายละเอียดจะได้กล่าวต่อไป

คำสั่ง writeln จะใช้ในการส่งข้อมูลออกทางจอภาพและเครื่องพิมพ์ โดยรายละเอียดของคำสั่งจะกล่าวต่อไป คำสั่งนี้จะนำข้อมูลที่อยู่ในเครื่องหมาย ‘ ‘ ออกทางหน้าจอ ถ้ามีข้อมูลหลายค่าจะใช้เครื่องหมาย , คั่น ถ้าหากข้อมูลที่จะพิมพ์อยู่ในเครื่องหมาย ‘ ‘ โปรแกรมจะมองว่าค่านั้นเป็นค่าที่เก็บอยู่ในตัวแปร ตัวอย่างเช่น

เอาต์พุต

```
WRITELN(' Hello. ');
```

Hello.

คำสั่งนี้อาจให้เอาต์พุตเป็นตัวแปร หรือการกระทำทางคณิตศาสตร์ก็ได้ ตัวอย่างเช่น

เอาต์พุต

```
x := 7;
```

```
WRITELN(x);
```

7

```
WRITELN(x + 9);
```

16

ถ้าหากคำสั่ง WRITELN มีทั้งข้อความและตัวแปร หรือการกระทำทางคณิตศาสตร์จะใช้เครื่องหมาย comma ; แยก โดยในโปรแกรมตัวอย่างจะเขียนคำสั่งเป็น

WRITELN(' Height in inches is ', inches)

เอาต์พุตที่ได้จะเป็นข้อความและตามด้วยค่าที่อยู่ในตัวแปร inches ดังนี้

Height in inches is 72

ถ้าหากโปรแกรมมีการทำงานตามคำสั่งต่อไปนี้


```
x := 8;
y := 6;
WRITELN(x, y);
WRITELN(x, ' ', y);
```

เอาต์พุตที่ได้จากการรันโปรแกรมจะเป็นดังนี้

86

8 6

คำสั่งที่ใช้ในการพิมพ์ข้อมูลอีกคำสั่งหนึ่งคือคำสั่ง WRITE โดยคำสั่งนี้หลังจากทำคำสั่งเคอร์เซอร์จะไม่ขึ้นบรรทัดใหม่ ส่วนคำสั่ง WRITELN หลังจากทำคำสั่งเคอร์เซอร์จะขึ้นบรรทัดใหม่

 **คำถาม** หลังจากคอมไพเลอร์ทำคำสั่งต่อไปนี้ เอาต์พุตที่ได้จะเป็นอย่างไร

```
yards := 8;
feet := yards * 3;
write(yards, ' yards is ');
write(feet, ' feet');
```

```
yards := 8;
feet := yards * 3;
writeln(yards, ' yards is ');
writeln(feet, ' feet');
```

คำตอบ

8 yards is24 feet

8 yards is

24 feet

 **คำถาม** ถ้าหากคอมพิวเตอร์ทำคำสั่งต่อไปนี้ เอาต์พุตที่ได้จะเป็นอย่างไร

```
Writeln ('Yes');
Writeln ('No');
```

```
Writeln('Yes');
Writeln;
Writeln('No');
```

```
Write('Yes');
Write('No');
```

คำตอบ


Yes

Yes

YesNo

No

No

 **คำถาม** เอาต์พุตทางจอภาพจะเป็นอย่างไร ถ้าหากคอมพิวเตอร์ทำโปรแกรมต่อไปนี้

```
Program NickelDime;
Var   nickels,dimes,TotCenter: Integer;
BEGIN
    Nickels := 3;
    dimes := 7;
    TotCents := (nickels * 5) + (dimes * 10);
    Writeln (nickels, ' nickels and ', dimes, ' dimes');
    Writeln ('= ',TotCents, ' cents')
END.
```

คำตอบ

3 nickels and 7 dimes

= 85 cents

3.3 การเก็บข้อมูลของตัวแปร

ในการประกาศตัวแปรให้กับโปรแกรม เมื่อโปรแกรมถูกรันตัวแปรต่าง ๆ จะเป็นตำแหน่งหน่วยความจำที่ใช้เก็บข้อมูล ตัวอย่างเช่นในโปรแกรม NickelDime จะใช้หน่วยความจำสามตำแหน่ง หลังจากโปรแกรมทำงานค่าในหน่วยความจำตำแหน่งต่าง ๆ จะเป็น

nickels 3	dimes 7	TotCents 85
--------------	------------	----------------

เมื่อโปรแกรมมีการทำงานใด ๆ ค่าในหน่วยความจำหรือตัวแปรอาจมีการเปลี่ยนแปลงได้ โดยค่าเก่าจะหายไปและถูกแทนด้วยค่าใหม่ พิจารณาโปรแกรมต่อไปนี้

```
PROGRAM drill
VAR   x: integer;

BEGIN
      x := 6;
      x := 8;
      WRITELN('X is ',x)
END.
```

หน่วยความจำ

x

6

8

เมื่อโปรแกรมมีการรัน เริ่มแรกค่าในตัวแปร x จะมีค่าเป็น 6 ต่อมามีการใส่ค่า 8 ลงในตัวแปร x ทำให้ค่าใน x เป็น 8 เมื่อโปรแกรมรันไปถึงคำสั่ง WRITELN จะทำให้พิมพ์เลข 8 ออกทางหน้าจอ

ในการเขียนโปรแกรมภาษา Pascal เราสามารถนำค่าในตัวแปรใส่ให้กับตัวแปรได้ โดยใช้เครื่องหมาย assignment (:=) เช่นเดียวกับการกำหนดค่าให้ตัวแปร พิจารณาโปรแกรมตัวอย่างต่อไปนี้

```
num1 := 5;
num2 := 14;
num1 := num2;
WRITELN(num1);
WRITELN(num2);
```

เอาต์พุต

14

14

เมื่อโปรแกรมทำงาน เริ่มแรกค่าในหน่วยความจำ num1 จะเป็น 5 และค่าใน num2 เป็น 14 ต่อมามีการใส่ค่าในตัวแปร num1 ด้วยค่า num2 ซึ่งจะทำให้ค่าในตัวแปร num1 เปลี่ยนไป เอาต์พุตจากการรันโปรแกรมจะได้เลข 14 จำนวนสองตัวทางจอภาพ

ในการกำหนดค่าให้กับตัวแปรนั้นเราสามารถนำค่าตัวแปรเก่ามากระทำทางคณิตศาสตร์ได้ โดยที่ตัวแปรทางซ้ายเปลี่ยนไปตามการกระทำทางขวามือ ดังตัวอย่างต่อไปนี้

- ก. `count := count + 1` {เพิ่มค่าในตัวแปร `count` ขึ้นหนึ่ง}
- ข. `sum := sum + x` {นำค่าใน `sum` บวกกับ `x` โดยค่าใน `x` จะไม่เปลี่ยน}
- ค. `num := 3 * num` {นำค่าใน `num` คูณกับ 3 และเก็บไว้ที่เดิม}

สำหรับการประกาศค่าคงที่จะใช้ในการกำหนดชื่อให้เป็นค่าคงที่ในโปรแกรม โดยค่านี้จะไม่เปลี่ยนแปลงตลอดการทำงานของโปรแกรม ค่าคงที่อาจเป็นข้อมูลชนิดตัวเลข หรือตัวอักขระก็ได้ การประกาศค่าคงที่จะเริ่มต้นด้วยคำว่า `CONST` ทำได้ 2 รูปแบบดังนี้

1. `CONST ชื่อ = ค่าคงที่ หรือสแตตเมนต์;`
2. `CONST ชื่อ : ชนิดข้อมูล = ค่าคงที่;`

ตัวอย่างเช่น

```
CONST Max = 500;
      Min : Integer = 100;
      Ave = (Max + Min) DIV 2;
```

บรรทัดแรกเป็นการกำหนดค่าคงที่คำว่า `Max` ให้แทนค่า 500 ซึ่งเป็นการกำหนดรูปแบบที่ 1 บรรทัดที่สองกำหนดให้ชื่อ `Min` เก็บค่าจำนวนเต็มมีค่าเท่ากับ 100 เป็นการกำหนดตามรูปแบบที่ 2 บรรทัดที่สามกำหนดได้ `Ave` มีค่าเท่ากับค่าเฉลี่ยระหว่าง 100 กับ 500 เป็นการกำหนดตามรูปแบบที่ 2

จากที่ผ่านมาเราจะพบว่า การกำหนดค่าให้ตัวแปรจะใช้เครื่องหมาย `:=` ซึ่งจะเป็นการกำหนดค่าในโปรแกรม แต่ถ้าต้องการจะกำหนดค่าให้กับตัวแปรโดยรับข้อมูลจากแป้นพิมพ์จะต้องใช้คำสั่งมาตรฐานในการรับข้อมูล ซึ่งมี 2 คำสั่งคือ `Read` และ `Readln` ทั้งสองคำสั่งจะเป็นการรับข้อมูลจากอุปกรณ์ภายนอก (โดยทั่วไปเป็นแป้นพิมพ์) เข้ามาเก็บในตัวแปรที่กำหนด โดยข้อมูลที่รับเข้ามาจะต้องสอดคล้องกับประเภทของตัวแปรที่ประกาศไว้ คำสั่ง `Read` และ `Readln` จะต่างกันตรงที่คำสั่ง `Read` เมื่อรับข้อมูลเข้าไปแล้วเคอร์เซอร์จะอยู่ที่เดิม ส่วนคำสั่ง `Readln` เมื่อรับข้อมูลเข้าไปแล้วเคอร์เซอร์จะขึ้นบรรทัดใหม่ แต่ในเทอร์โบปาสคาลรุ่นใหม่ ๆ ทั้งสองคำสั่งจะทำหน้าที่เหมือนกัน รูปแบบการใช้คำสั่ง `Read` และ `Readln` เป็นดังนี้

```
Read(var1 [,var2][,var3][,.....,varN]);
Readln(var1 [,var2][,var3][,.....,varN]);
```


การใช้คำสั่ง Read และ Readln มีหลายรูปแบบ ตัวแปรที่เก็บข้อมูลจะต้องอยู่ในวงเล็บที่ตามหลังคำสั่ง โดยอาจเก็บข้อมูลหลาย ๆ ตัวแปรหรือเก็บในตัวแปรเดียวก็ได้ แต่ถ้าหากใช้คำสั่งที่ไม่มีวงเล็บตามหลัง จะให้ผู้ใช้โปรแกรมป้อนอะไรก็ได้จนกว่าจะกดคีย์ Enter โดยจะไม่มีกรเก็บค่าที่ป้อนไว้ในตัวแปร เมื่อโปรแกรมทำงานมาถึงคำสั่ง Read หรือ Readln คอมพิวเตอร์จะหยุดรอการป้อนข้อมูล ถ้าหากเป็นคำสั่ง Read หรือ Readln ที่รับข้อมูลเข้าไปหลายตัวแปรจะต้องกดคีย์เว้นวรรคระหว่างตัวแปรแต่ละตัว ตัวอย่างการใช้คำสั่งแสดงได้ดังตารางต่อไปนี้

คำสั่ง	ข้อมูลอินพุต	ผลที่ได้หลังจากทำคำสั่ง
1. Read(X)	32	X = 32
2. Read(A,B,C)	3 4 60	A = 3,B = 4,C = 60
3. Read(D,E)	24 (กด Enter) 76	D = 24,E = 76
4. Read(Y,Z)	46 32.4 15	Y = 46,Z = 32,15 จะไม่ถูกเก็บ

จากการทำงานของคำสั่งในตารางตัวแปร Z จะต้องประกาศเป็น Real ส่วนตัวแปรอื่น ๆ จะประกาศเป็น Integer จะสังเกตเห็นว่าในคำสั่งที่ 4 จะไม่มีการเก็บค่า 15 เนื่องจากในคำสั่งระบุให้รับตัวแปรเข้าไปเก็บจำนวน 2 ตัวเท่านั้น

สำหรับการใช้คำสั่ง Readln เมื่อรับข้อมูลไปแล้วเคอร์เซอร์จะไปอยู่ในบรรทัดใหม่ และข้อมูลจะถูกเก็บในคำสั่งเมื่อกดคีย์ Enter ถ้าหากมีการป้อนข้อมูลต่อไปนี้โดยการป้อนแต่ละบรรทัดจบด้วยการกดคีย์ Enter

```
10 20
15 16
22 21
```

ในตารางต่อไปจะแสดงการทำงานของคำสั่ง Readln หลังจากการป้อนข้อมูลเข้าไปข้างต้น โดยแสดงว่าถ้าหากเขียนคำสั่งในลักษณะของสเตตเมนต์ต่าง ๆ และมีการป้อนข้อมูลเข้าไปตามรูปแบบข้างต้น ค่าที่เก็บในตัวแปรจะเป็นอย่างไร

สเตตเมนต์	ผลลัพธ์หลังจากทำคำสั่งของสเตตเมนต์
1. Readln(A); Readln(B); Readln(C);	A = 10 B = 15 C = 22

2. Readln(A,B,C); Readln(D,E);	A = 10,B = 20,C = 15 D = 22,E = 21
3. Readln(A,B); Readln(C,D); Readln(E);	A = 10,B = 20 C = 15,D = 16 E = 22
4. Readln(A); Readln(B); Readln(C,D,E);	A = 10 B = 15 C = 22,D = 21 , E จะ Error ไม่มีข้อมูลเก็บ
5. Readln(A,B); Readln; Readln(C);	A = 10,B = 20 C = 22

จากตารางจะเห็นว่าในสแตตเมนต์ที่ 1 คำสั่งแรกจะรับข้อมูลมาเก็บในตัวแปร A ตัวเดียว แต่เราอินพุต 10 20 และกด Enter ทำให้ตัวแปร A เก็บค่า 10 ส่วน 20 จะถูกตัดทิ้งไป สแตตเมนต์ที่ 2 คำสั่งแรกจะรับข้อมูลเก็บได้ 3 ตัว แต่เราป้อนข้อมูล 2 ตัวคือ 10 20 และกด Enter ทำให้ตัวแปร A เก็บ 10 ตัวแปร B เก็บ 20 และเหลือตัวแปรอีกหนึ่งตัวคือ C เมื่อป้อน 15 และ 16 ในบรรทัดต่อไปทำให้ 15 ถูกเก็บใน C ส่วน 16 จะถูกตัดทิ้ง ส่วนสแตตเมนต์อื่น ๆ พิจารณาได้ในทำนองเดียวกัน

ถ้าหากในโปรแกรมมีการเขียนสแตตเมนต์ต่อไปนี้

```

VAR    a,b,c,d,e : Integer;
.....
.....
Readln(a,b);
Read(c,d);
Read(e);
.....
    
```

เมื่อรันโปรแกรมมีการป้อนข้อมูลดังนี้

11 12 13 14 (กด Enter)

15 16 17 18 (กด Enter)

ผลลัพธ์ที่ได้จะเป็น

$$a = 11, b = 12, c = 15, d = 16, e = 17$$

3.4 โปรแกรมอินเตอร์แอกทีฟ (Interactive Program)

โปรแกรมคอมพิวเตอร์ที่มีการตอบสนองการทำงานภายหลังที่ผู้ใช้ทำการป้อนค่าอินพุตเข้าไปจะเรียกว่าโปรแกรมอินเตอร์แอกทีฟ (interactive) ตัวอย่างต่อไปเป็นโปรแกรมที่คอมพิวเตอร์จะให้ป้อนค่าข้อมูลที่เป็นค่าฟุต จากนั้นคอมพิวเตอร์จะคำนวณค่าในหน่วยของนิ้วและแสดงทางหน้าจอ

```
PROGRAM FeetToInches;
VAR   feet,inches : integer;
BEGIN
      write('Enter number of feet ');
      readln(feet);
      inches := feet * 12;
      writeln(inches, ' inches ')
END.
```

เมื่อรันโปรแกรม คอมพิวเตอร์จะแสดงเครื่องหมายพร้อม(Prompts) ทางหน้าจอ พร้อมกับข้อความ

Enter number of feet __

จากนั้นเมื่อใส่ข้อมูลเข้าไปเครื่องจะคำนวณค่าในหน่วยนิ้วออกมา จะเห็นว่าโปรแกรมในลักษณะนี้จะต้องมีการใช้คำสั่งสำหรับให้ผู้ใช้ติดต่อกับเครื่อง ในโปรแกรมนี้อาจจะใช้คำสั่ง readln ซึ่งจะได้รับค่าข้อมูลทางแป้นพิมพ์ เมื่อคอมพิวเตอร์ทำงานมาถึงคำสั่งนี้คอมพิวเตอร์จะแสดงเครื่องหมาย Prompt ถ้าหากมีการใส่ข้อมูลใด ๆ เข้ามาและกด Enter ข้อมูลนั้นจะถูกนำไปเก็บในตัวแปรที่อยู่ใต้วงเล็บตามหลักคำสั่ง readln

ในโปรแกรมแบบ interactive การเขียนข้อความเพื่อใช้ร่วมกับเครื่องหมาย Prompt ที่เกิดจากคำสั่ง readln มักจะใช้คำสั่ง write ในการแสดงข้อมูล เพื่อให้เครื่องหมาย Prompt อยู่ต่อจากข้อความที่ปรากฏ ถ้าหากใช้คำสั่ง writeln เครื่องหมาย Prompt จะอยู่อีกบรรทัดหนึ่ง เราอาจสังเกตได้ว่าโปรแกรมส่วนใหญ่คำสั่ง readln มักจะตามหลังคำสั่ง write

แบบฝึกหัด

1. จงเติมส่วนต่าง ๆ ลงไปในโปรแกรมต่อไปนี้ให้สมบูรณ์

```
PROGRAM Prog8
VAR feet,inches integer
BEGIN
    feet := 5
    inches := feet * 12
    WRITELN(inches)
END
```

2. เอาต์พุตของส่วนของโปรแกรมต่อไปนี้จะเป็นอย่างไร

```
x := 5;
y := 8;
writeln('x equals ',x);
writeln(x,y);
writeln(x, ' equals x');
```

3. จงเขียนโปรแกรมต่อไปนี้ให้สมบูรณ์ และตอบว่าเอาต์พุตของโปรแกรมจะเป็นอย่างไร

ก.

```
num := 10;
x := 5;
write(num);
write(x);
writeln(num);
```

ค.

```
x := 4;
y := 6;
y := y + x;
x := x + 1;
writeln('x = ',x);
writeln('y = ',y);
```

ข.

```
numb := 10;
x := 5;
write(numb, ' ');
write(x, ' ');
writeln;
writeln(numb);
```

ง.

```
numb1 := 5;
numb2 := 9;
numb1 := numb1 + 1;
numb2 := numb2 + numb1;
writeln('numb1 = ',numb1);
writeln('numb2 = ',numb2);
```