

ส่วนที่ 2

แบบฝึกหัดท้ายบท

แบบฝึกหัดท้ายบทที่ 2

แบบฝึกหัดท้ายบทที่ 2 ข้อ 1

จงเขียนโปรแกรมภาษาปาสคาลเพื่อหาค่ากำลังสอง กำลังสาม และกำลังสี่ ของเลขใดๆ

```
PROGRAM EX1;
VAR A , A2 , A3 , A4 : integer;           ... (1)
BEGIN
    WRITE('PLEASE PUT 1 DATA'); READLN(A); ... (2)
    A2 := SQR(A);                          ... (3)
    A3 := A * A2 ;                          ... (4)
    A4 := SQR(SQR(A)) ;                     ... (5)
    WRITELN('A^2=' , A2);                   ... (6)
    WRITELN('A^3=' , A3);
    WRITELN('A^4=' , A4);
END.
```

คำอธิบายโปรแกรม

- (1) เป็นการประกาศตัวแปรที่ใช้ทั้งหมดในโปรแกรม โดยตัวเลขที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์ จะนำมาเก็บในตัวแปร A ส่วน ตัวแปร A2 , A3 , A4 เก็บค่าของกำลังสอง กำลังสาม และกำลังสี่ที่เป็นผลลัพธ์จากการทำงาน
- (2) เป็นคำสั่งในการรับเลขจำนวนเต็ม 1 จำนวนทางแป้นพิมพ์โดยหน้าจจะปรากฏข้อความ ดังนี้ PLEASE PUT 1 DATA ?

- (3) เป็นคำสั่งเรียกฟังก์ชัน SQR ซึ่งเป็นการหาค่ากำลังสองของ A นำผลจากการคำนวณให้ค่าแก่ตัวแปร A2
- (4) เป็นการคำนวณหาค่ากำลังสามให้ค่าแก่ตัวแปร A3
- (5) เป็นการเรียกฟังก์ชัน SQR ซ้อน อีกครั้งหนึ่งซึ่งคือค่าของ A ยกกำลังสี่นั่นเอง นำผลจากการคำนวณให้ค่าแก่ตัวแปร A4
- (6) เป็นคำสั่งในการแสดงผลลัพธ์จากการทำงานสมมุติว่าผู้ใช้ป้อนค่า 3 ผลของการทำงานเป็นดังนี้

$$A^2 = 9$$

$$A^3 = 27$$

$$A^4 = 81$$

แบบฝึกหัดท้ายบทที่ 2 ข้อ 2

จงเขียนโปรแกรมภาษาปาสคาลเพื่อหาค่าเฉลี่ยของเลขสามจำนวนใดๆ

PROGRAM EX2;

VAR A , B , C :INTEGER; ... (1)

AVERAGE : REAL ; ... (2)

BEGIN

WRITE('PLEASE PUT 3 NUMBERS'); READLN(A, B, C); ... (3)

AVERAGE := (A+B+C) / 3 ; ... (4)

WRITELN('AVERAGE =', AVERAGE:5:2); ... (5)

END.

คำอธิบายโปรแกรม

- (1) เป็นการประกาศตัวแปร A , B, C เป็นเลขจำนวนเต็มโดยตัวแปรทั้งสามนี้จะรับค่าจากผู้ใช้ที่ป้อนทางแป้นพิมพ์ 3 จำนวนด้วยกัน
- (2) เป็นการประกาศตัวแปร AVERAGE เพื่อเก็บผลลัพธ์จากการคำนวณซึ่งคือค่าเฉลี่ยที่โจทย์ต้องการนั่นเอง

- (3) เป็นคำสั่งในการรับค่าซึ่งเป็นเลขจำนวนเต็มทางแป้นพิมพ์สามจำนวน โดยมีข้อความปรากฏทางจอภาพดังนี้ PLEASE PUT 3 NUMBERS ต่อจากนั้นผู้ใช้จะป้อนตัวเลขจำนวนเต็ม 3 จำนวนทางแป้นพิมพ์ สมมุติว่าป้อน 3 4 5 ค่าทั้งสามค่านี้จะเก็บในตัวแปร A , B และ C ตามลำดับ
- (4) เป็นคำสั่งในการนำค่าของผลรวมของ A, B ,C ทหารด้วย 3 เก็บในตัวแปร AVERAGE
- (5) แสดงผลออกทางจอภาพดังนี้ AVERAGE = 3.00 โดยกำหนดทศนิยม 2 ตำแหน่ง

แบบฝึกหัดท้ายบทที่ 2 ข้อ 3

จงเขียนโปรแกรมคำนวณหารายได้ต่อปีของบุคคลใดๆ โดยป้อนเงินเดือน ทางแป้นพิมพ์ นำรายได้มาคิดหักภาษีดังนี้ นำรายได้ทั้งปีหักค่าใช้จ่ายส่วนตัว 40% และหักค่าลดหย่อนอีก 30000 บาท

PROGRAM EX3;

VAR SALARY ,YEAR , NET : REAL ; ... (1)

BEGIN

WRITE('SALARY = ? '); READLN(SALARY); ... (2)

YEAR := 12 * SALARY ; ... (3)

NET := YEAR - (0.4*YEAR) - 30000 ; ... (4)

WRITELN('NET = ' , NET :6:2) ... (5)

END.

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปรที่ใช้ในโปรแกรมโดย SALARY หมายถึงเงินเดือน YEAR หมายถึงเงินได้ทั้งปี และ ตัวแปร NET เป็นตัวแปรที่เก็บผลลัพธ์ซึ่งคือ เงินได้สุทธิที่หักค่าใช้จ่ายส่วนตัว และหักค่าลดหย่อนแล้ว
- (2) เป็นการรับเงินเดือนทางแป้นพิมพ์ หน้าจอจะแสดงข้อความ SALARY = ?
- (3) คำนวณหารายได้ต่อปีโดยนำเงินเดือนคูณด้วย 12 เก็บในตัวแปร YEAR
- (4) นำรายได้ทั้งปี หักค่าใช้จ่ายส่วนตัว 40% และหักค่าลดหย่อนอีก 30000 ค่ารวมเก็บใน

ตัวแปร NET

- (5) พิมพ์ผลลัพธ์ออกทางจอภาพ กำหนดทศนิยม 2 ตำแหน่ง

แบบฝึกหัดท้ายบทที่ 3

แบบฝึกหัดท้ายบทที่ 3 ข้อ 1

จงเขียนโปรแกรมในการแปลงจำนวนหลา จำนวนฟุต และจำนวนนิ้วใดๆให้เป็นจำนวนนิ้วทั้งหมด

PROGRAM EX4;

VAR YARD ,FEET , INCH , TOTAL : INTEGER ; ... (1)

BEGIN

WRITE('YARD='); READLN(YARD); ... (2)

WRITE('FEET='); READLN(FEET); ... (3)

WRITE('INCH'); READLN(INCH); ... (4)

TOTAL := YARD*36 + FEET*12 + INCH ; ... (5)

WRITELN('TOTAL OF INCH =', TOTAL); ... (6)

END.

คำอธิบายโปรแกรม

- (1) เป็นการประกาศตัวแปรที่ใช้ในโปรแกรม ตัวแปร YARD หมายถึงจำนวนหลา ตัวแปร FEET หมายถึงจำนวนฟุต ตัวแปร INCH หมายถึงจำนวนนิ้ว และ TOTAL เป็นตัวแปรที่เก็บผลลัพธ์จากการแปลงจำนวนหลา จำนวนฟุต และจำนวนนิ้วให้เป็นจำนวนนิ้วรวม
- (2) - (3) - (4) เป็นการรับค่าหลา ฟุต และนิ้วใดๆทางแป้นพิมพ์
- (5) เป็นคำสั่งในการคำนวณค่าโดยการแปลงค่า ซึ่ง 12 นิ้วมีค่าเท่ากับ 1 ฟุต และ 36 ฟุตมีค่าเท่ากับ 1 หลา ดังนั้น 1 หลา จึงมีค่าเท่ากับ 36 นิ้ว คำนวณและเก็บในตัวแปรชื่อ TOTAL
- (6) พิมพ์ผลลัพธ์จากการคำนวณพิมพ์ออกมาทางจอภาพ

แบบฝึกหัดท้ายบทที่ 3 ข้อ 2

จงเขียนโปรแกรมเพื่อหาจำนวนแบงค์ และจำนวนเหรียญที่น้อยที่สุด โดยป้อนจำนวนเงินที่เป็นเลขจำนวนเต็มใดๆทางแป้นพิมพ์ 1 จำนวน

PROGRAM EX5;

```
VAR   Money : integer ;                               ... (1)
      B1000, B500, B100, B50, B20 , C10 ,C5 ,C1 : INTEGER; ... (2)
BEGIN
    WRITE('money to exchange ='); READLN(Money);    ... (3)
    B1000 := Money div 1000 ;                          ... (4)
    B500  := Money mod 1000 div 500 ;                  ... (5)
    B100  := Money mod 1000 mod 500 div 100 ;         ... (6)
    B50   := Money mod 1000 mod 500 mod 100 div 50 ;  ... (7)
    B20   := Money mod 1000 mod 500 mod 100 mod 50 div 20 ; ... (8)
    C10   := Money mod 1000 mod 500 mod 100 mod 50 mod 20 div 10 ;
    C5    := Money mod 1000 mod 500 mod 100 mod 50 mod 20 mod 10 div 5 ;
    C1    := Money mod 1000 mod 500 mod 100 mod 50 mod 20 mod 10 mod 5 ;
    WRITELN(' Bank of 1000 =',B1000);                 ... (9)
    WRITELN(' Bank of 500 =',B500);
    WRITELN(' Bank of 100 =',B100);
    WRITELN(' Bank of 20 =',B20);
    WRITELN(' Bank of 10 =',B10);
    WRITELN(' coin of 10 =',C10);
    WRITELN(' coin of 5 =',C5);
    WRITELN(' coin of 1 =',C1);
END.
```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร Money เป็นเลขจำนวนเต็ม สำหรับเก็บค่าของจำนวนเงินที่ผู้ใช้ป้อน
- (2) ประกาศตัวแปรที่เก็บจำนวนของแบงค์ 1000 แบงค์ 500 แบงค์ 100 แบงค์ 50 แบงค์ 20 เหรียญ 10 เหรียญ 5 และเหรียญ 1 บาท ซึ่งเป็นผลลัพธ์จากการปฏิบัติงานในโปรแกรม
- (3) เป็นคำสั่งในการรับเลขจำนวนเต็ม 1 จำนวนทางแป้นพิมพ์ ซึ่งคือจำนวนเงินที่ผู้ใช้ต้องการแลกให้เป็นแบงค์และเหรียญที่น้อยที่สุด นำมาเก็บในตัวแปร Money
- (4) นำจำนวนเงินมาหารด้วย 1000 โดยใช้ตัวกระทำ div ให้ผลลัพธ์เป็นเลขจำนวนเต็มซึ่งหมายถึงจำนวนแบงค์ 1000 ที่สามารถแลกได้
- (5) นำเศษที่เหลือในที่นี้ต้องมีค่าไม่เกิน 1000 มาหารด้วย 500 ผลที่ได้มี 2 ค่าคือ 0 ในกรณีที่มีเศษมีค่าไม่ถึง 500 และค่า 1 กรณีที่มีเศษมีค่าเกิน 500
- (6) นำเศษที่ได้จากข้อ 5 ซึ่งมีค่าไม่เกิน 500 มาหารด้วย 100 ผลที่ได้คือจำนวนของแบงค์ 100 ที่แลกได้นั่นเอง ซึ่งค่าที่ได้มีค่าในช่วง 0 ถึง 4
- (7) นำเศษที่ได้จากข้อ 6 มาหารด้วย 50 ผลที่ได้คือจำนวนของแบงค์ 50 ที่สามารถแลกได้
- (8) นำเศษที่ได้ในข้อ 7 มาหารด้วย 20 ผลที่ได้คือจำนวนของแบงค์ 20 ที่สามารถแลกได้ ต่อจากนั้นทำในลักษณะเดียวกันกับการทำงานข้างต้นโดยนำเศษที่ได้มาหาร กับค่าของเหรียญ 10 เหรียญ 5 และเหรียญบาท เพื่อหาจำนวนเหรียญที่สามารถแลกได้จำนวนน้อยที่สุด
- (9) แสดงผลที่ได้ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 3 ข้อ 3

จงเขียนโปรแกรมภาษาปาสคาลหาพื้นที่ของสามเหลี่ยมใดๆโดย

สูตรในการหาพื้นที่คือ $\text{Sqrt}(s*(s-a)*(s-b)*(s-c))$

โดยที่ S คือ $(a+b+c) / 2$

และ a, b, c คือความยาวด้านทั้งสามของสามเหลี่ยม

```

PROGRAM EX6;
VAR  AREA , S : REAL;           ... (1)
     A , B, C  : INTEGER;       ... (2)
BEGIN
    WRITE('PLEASE PUT 3 INTEGER NUMBER'); READLN(A,B,C);   ... (3)
    S := (A+B+C) / 2 ;          ... (4)
    AREA := SQRT(S*(S-A)*(S-B)*(S-C));  ... (5)
    WRITELN('AREA = ' , AREA:6:2);     ... (6)
END.

```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร AREA และ S เป็นเลขทศนิยม
- (2) ประกาศความยาวด้านทั้งสามของสามเหลี่ยมเป็นชนิดจำนวนเต็ม
- (3) รับเลขจำนวนเต็ม 3 จำนวนทางแป้นพิมพ์เป็นความยาวด้านทั้งสามของสามเหลี่ยม
- (4) คำนวณหาค่าของ S
- (5) คำนวณหาพื้นที่โดยใช้ฟังก์ชัน SQRT คือการหาค่ารากที่สองนั่นเอง
- (6) พิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 4

แบบฝึกหัดท้ายบทที่ 4 ข้อ 1

จงเขียนโปรแกรมภาษาปาสคาลโดยรับข้อมูลเป็นเลขจำนวนเต็ม 2 จำนวนทางแป้นพิมพ์ ถ้าเลขสองจำนวนนั้นเป็นเลขจำนวนเต็มคู่ หรือเลขจำนวนเต็มคี่ อย่างละหนึ่งจำนวน ให้พิมพ์ TRUE ถ้าเป็นเลขคี่ทั้งสองจำนวนหรือเลขคู่ทั้งสองจำนวนให้พิมพ์ False

```

PROGRAM EX7;
VAR  A, B : INTEGER;                ... (1)
     Check_ODD, Check_EVEN , Check : BOOLEAN;    ... (2)
BEGIN
    WRITE("Please put 2 integer"); READLN(A,B);    ... (3)
    Check_ODD := Odd(A) and Odd(B);              ... (4)
    Check_EVEN := (A mod 2 = 0) and ( B mod 2 = 0 );    ... (5)
    Check      := not (Check_ODD or Check_EVEN);    ... (6)
    WEITELN(Check);                              ... (7)
END.

```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร A และ B เป็นเลขจำนวนเต็ม
- (2) ประกาศตัวแปร Check_ODD เพื่อตรวจสอบข้อมูลว่าตัวเลขทั้งสองจำนวนเป็นเลขจำนวนเต็มคือ ตัวแปร Check_EVEN เป็นตัวแปรตรวจสอบว่าเลขจำนวนเต็มทั้งสองจำนวนเป็นเลขจำนวนเต็มคู่ทั้งสองจำนวน และตัวแปร Check เป็นตัวแปรที่เก็บผลลัพธ์ของการตรวจสอบว่าเป็นเลขจำนวนเต็มคู่และจำนวนเต็มคี่อย่างละ 1 จำนวน
- (3) รับเลขจำนวนเต็มใดๆ 2 จำนวนทางแป้นพิมพ์
- (4) เป็นนิพจน์ของการตรวจสอบว่าเลขจำนวนเต็มทั้งสองจำนวนเป็นเลขจำนวนเต็มคี่ทั้งสองจำนวนหรือไม่ โดยใช้ฟังก์ชัน odd กรณีที่เป็นตัวเลขเป็นเลขจำนวนเต็มคี่ให้ผลลัพธ์เป็น TRUE
- (5) เป็นนิพจน์ของการตรวจสอบว่าเลขจำนวนเต็มทั้งสองเป็นเลขจำนวนเต็มคู่ทั้งสองจำนวนหรือไม่ ตรวจสอบโดยนำ 2 มาหารเลขจำนวนเต็มนั้นถ้าหารแล้วได้เศษเท่ากับ 0 แสดงว่าหารลงตัว ดังนั้นตัวเลขที่ทำการตรวจสอบเป็นเลขจำนวนเต็มคู่นั่นเอง
- (6) เป็นนิพจน์ที่ตรวจสอบให้ค่าเป็น TRUE ถ้าตัวเลขจำนวนเต็ม 2 ตัวที่ตรวจสอบ ไม่เป็นเลขจำนวนเต็มคี่ทั้งสองจำนวน หรือไม่เป็นเลขจำนวนเต็มคู่ทั้งสองจำนวน
- (7) พิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 4 ข้อ 2

จงเขียนโปรแกรมภาษาปาสคาลตรวจสอบปี คศ.ที่ป้อนทางแป้นพิมพ์ว่าเป็นปีอธิกวารหรือไม่ ปีอธิกวารคือปีที่มี 366 วัน ปกติ 4 ปีจะมี 1 ครั้งสังเกตได้ว่าปี คศ.ใดที่หารด้วย 4 ลงตัวจะเป็นปีที่มี 366 วัน เช่น ปีคศ. 1992 1993 1994 1995 1996 ปีที่มี 366 วันคือปี คศ. 1992 และ 1996 แต่ถ้าเป็นปีคศ.ที่ลงท้ายด้วย 00 เช่นปีคศ. 1200 1300 1400 1500 1600 จะเห็นว่าหารด้วย 4 ลงตัวหมดแต่ปีที่มี 366 วันจริงๆคือปีคศ. 1200 และปีคศ. 1600 ซึ่งหารด้วย 400 ลงตัวเท่านั้น

PROGRAM EX8;

VAR LeapYear : BOOLEAN ; ... (1)

Year : INTEGER ; ... (2)

BEGIN

WRITE('YEAR='); READLN(Year); ... (3)

LeapYear := (Year mod 4 = 0) and (Year mod 100 <> 0) or ... (4)

(Year mod 400 = 0);

WRITELN(LeapYear); ... (5)

END.

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร LeapYear เป็นตัวแปรชนิดบูลีนเพื่อตรวจสอบค่าปีว่ามี 366 วันหรือไม่
- (2) ประกาศตัวแปร Year เป็นเลขจำนวนเต็มเพื่อเก็บปี คศ.ใดๆที่ต้องการตรวจสอบ
- (3) ป้อนค่าปีใดๆทางแป้นพิมพ์
- (4) ตรวจสอบโดยใช้นิพจน์บูลีน แบ่งการตรวจสอบเป็น 2 ลักษณะคือปีที่ไม่ได้ลงท้ายด้วย 00 แสดงว่าเป็นปีที่หาร 100 แล้วมีค่าไม่เป็นศูนย์ ต้องหาร 4 ลงตัวเท่านั้นถึงจะเป็นปีอธิกวาร อีกกรณีหนึ่งคือปีที่หารด้วย 100 ลงตัวจะต้องหารด้วย 400 ลงตัวด้วยจึงเป็นปีอธิกวาร ทั้งสองกรณีนี้กรณีใดเป็นจริง ก็มีผลเป็นจริง จึงใช้ or เป็นตัวกระทำระหว่างนิพจน์
- (5) พิมพ์ค่าผลลัพธ์ออกทางจอภาพ ค่าที่พิมพ์มีค่าไม่ TRUE ก็ FALSE ค่าใดค่าหนึ่งเท่านั้น

แบบฝึกหัดท้ายบทที่ 5

แบบฝึกหัดข้อ 1 หน้า 72

จงเขียนโปรแกรมภาษาปาสคาลเพื่อตัดเกรดของนักศึกษา ตามข้อมูลดังนี้

คะแนนที่สอบได้	เกรดที่ได้
80 – 100	A
70 – 79	B
60 – 69	C
50 – 59	D
น้อยกว่า 50	F

Programtest1;

Var

Score : integer;

Begin

Write ('Input score of student = ');

readln (score); ... (1)

if score > 79 then writeln ('A') ... (2)

else if score > 69 then writeln ('B')

else if score > 59 then writeln ('C')

else if score > 49 then writeln ('D')

else writeln ('F'); ... (3)

End.

คำอธิบายโปรแกรม

- (1) รับคะแนนของนักศึกษาทางแป้นพิมพ์เป็นเลขจำนวนเต็ม 1 จำนวน
- (2) ตัดเกรดโดยใช้คำสั่งเงื่อนไข ตามเงื่อนไขที่โจทย์กำหนด
- (3) จุดสิ้นสุดของคำสั่งเงื่อนไข

แบบฝึกหัดข้อ 2 หน้า 72

การคิดค่าไฟฟ้าของคอนโดมิเนียมแห่งหนึ่งมีอัตราการคิดค่าไฟฟ้าดังนี้

จำนวนหน่วยที่ใช้	อัตราต่อหน่วย (บาท)
0 – 100	3 บาท
101 – 500	100 หน่วยแรกคิดหน่วยละ 3 บาท ส่วนที่เกินคิดหน่วยละ 3 บาท
มากกว่า 500	100 หน่วยแรกคิดหน่วยละ 3 บาท ส่วนที่เกินแต่ไม่ถึง 501 คิดหน่วยละ 4 บาท ส่วนที่เกินคิดหน่วยละ 5 บาท

จงเขียนโปรแกรมรับหมายเลขจำนวนไฟฟ้าเดือนที่แล้ว และหมายเลขจำนวนใช้ไฟฟ้าของเดือนนี้ เพื่อคำนวณหาค่าไฟฟ้าของคอนโดมิเนียมนี้

```
Programtest2;
```

```
Var
```

```
Vol, Payment : integer;
```

```
VLmonth, VTmonth : integer;
```

```
Begin
```

```
Write('กรุณาป้อนหมายเลขไฟฟ้าเดือนที่แล้ว'); readln (VLmonth) ... (1)
```

```
Write('กรุณาป้อนหมายเลขไฟฟ้าเดือนนี้'); readln (VTmonth);
```

```
Vol := VTmonth - VLmonth;           {จำนวนหน่วยที่ใช้} ... (2)
```

```
If Vol <= 100 then Payment := Vol * 3 ... (3)
```

```
else if Vol < 501 then Payment := (100*3) + (Vol - 100) * 4
```

```
else Payment := (100*3) + (400*4) + (Vol-500)*5;
```

```
writeln ('ท่านใช้ไฟฟ้าเท่ากับ, Vol , 'หน่วย'); ... (4)
```

```
writeln('เป็นจำนวนเงินทั้งหมดเท่ากับ', Payment, 'บาท');
```

```
End.
```

คำอธิบายโปรแกรม

- (1) รับหมายเลขไฟฟ้าที่ใช้ของเดือนที่แล้วและเดือนนี้ทางแป้นพิมพ์
- (2) หาจำนวนหน่วยที่ใช้โดยนำหมายเลขไฟฟ้าของเดือนนี้ลบด้วยหมายเลขไฟฟ้าของเดือนที่แล้ว เก็บในตัวแปร Vol
- (3) คำสั่งเงื่อนไขเพื่อทำงานตามที่โจทย์ต้องการ
- (4) พิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 5 ข้อ 1

จงเขียนโปรแกรมภาษาปาสคาลอ่านค่าตัวเลขจำนวนเต็มทางแป้นพิมพ์ จำนวนหนึ่ง และจงหาค่าที่มากที่สุด และน้อยที่สุดของเลขชุดนี้คือ

1.1 เมื่อจำนวนของข้อมูลทั้งหมดบอกไว้ที่ตัวเลขตัวแรก

```
Program      test5_1;
Var
    N, Max, Min , size : integer;                .....(1)
begin
    write('จำนวนข้อมูลทั้งหมด'); readln (size) ;    .....(2)
    readln(N);          {อ่านข้อมูลตัวแรกทางแป้นพิมพ์}    .....(3)
    Max := N; Min := N;  {ให้ข้อมูลตัวแรกเป็นค่าสูงที่สุดและค่าที่ต่ำที่สุด}
    for I := 2 to size do    .....(4)
        begin
            readln(N);      {อ่านข้อมูลตัวต่อไป}
            if Max<N then Max := N;    .....(5)
            if Min>N then Min := N;    .....(6)
        end;
    writeln ('ข้อมูลที่น้อยที่สุดคือ',Min)    .....(7)
    writeln ('ข้อมูลทีมากที่สุดคือ', Max) ;
End.
```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปรทั้งหมดเป็นเลขจำนวนเต็ม
- (2) รับจำนวนข้อมูลทั้งหมดทางแป้นพิมพ์เก็บในตัวแปร Size
- (3) รับข้อมูลทั้งหมดตัวแรก
- (4) ทำจำนวนรอบเท่ากับ Size-1 รอบ
- (5) ทำการเปรียบเทียบค่าเพื่อนำค่าที่มากที่สุดเก็บในตัวแปร Max
- (6) เปรียบเทียบค่าเพื่อนำค่าที่น้อยที่สุดเก็บในตัวแปร Min
- (7) พิมพ์ผลลัพธ์ออกทางจอภาพ

1.2 เมื่อตรวจสอบหมดข้อมูลด้วยเลขที่ติดลบ

```
Program      test5_2;
Var
    N, Min, Max : integer;
Begin
    read (N); .....(1)
    Max := N; Min := N; .....(2)
    while N > 0 do .....(3)
        begin
            if N < Min then Min := N; .....(4)
            if N > Max then Max := N; .....(5)
            readln (N); .....(6)
        end;
        writeln ('ข้อมูลที่น้อยที่สุดคือ', Min) .....(7)
        writeln ('ข้อมูลที่มากที่สุดคือ', Max) ;
    End.
```

คำอธิบายโปรแกรม

- (1) อ่านข้อมูลตัวแรกเก็บในตัวแปร N
- (2) ให้ข้อมูลตัวแรกเป็นทั้งค่ามากที่สุดและค่าน้อยที่สุด
- (3) ทำงานวนรอบถ้าข้อมูล N มีค่ามากกว่า 0 กรณีที่เป็นเท็จจะหยุดการทำงานวนรอบ
- (4) เป็นการตรวจสอบเงื่อนไขเพื่อนำค่ามากที่สุดเก็บในตัวแปร Max
- (5) ทำการตรวจสอบเงื่อนไขเพื่อนำค่าน้อยที่สุดเก็บในตัวแปร Min
- (6) อ่านข้อมูลตัวถัดไปเก็บในตัวแปร N
- (7) พิมพ์ผลลัพธ์ออกทางจอภาพ

1.3 ใช้ EOF

```
Program test5_3;
Var
    N, Min, Max : integer;
begin
    Max := -Maxint ; Min := Maxint ; .....(1)
    While NOT EOF do .....(2)
        begin
            read (N); .....(3)
            if N < Min then Min := N; .....(4)
            if N > Max then Max := N; .....(5)
        end;
        writeln ('ข้อมูลที่น้อยที่สุดคือ', Min) .....(6)
        writeln ('ข้อมูลที่มากที่สุดคือ', Max) ;
    End.
```

คำอธิบายโปรแกรม

- (1) กำหนดให้ตัวแปร Max มีค่าเท่ากับ -32768 และ ตัวแปร Min มีค่าเท่ากับ 32767 ซึ่งกำหนดค่าเริ่มต้นให้กับตัวแปรทั้งสอง
- (2) ในกรณีที่ยังไม่พบ CTRL-Z หรือจุดสิ้นสุดของข้อมูลจะกระทำงานวนรอบไปเรื่อยๆ
- (3) ในกรณีที่ยังไม่หมดข้อมูลจะทำการอ่านข้อมูลเป็นเลขจำนวนเต็มทางแป้นพิมพ์
- (4) นำค่าที่ป้อนมาเปรียบเทียบกับค่าของตัวแปร Min เพื่อนำค่าที่น้อยกว่าเก็บในตัวแปร Min
- (5) ในทำนองเดียวกัน นำค่าที่ป้อนมาเปรียบเทียบกับค่าของตัวแปร Max เพื่อนำค่าที่มากกว่าไปเก็บในตัวแปร Max
- (6) ในกรณีที่หมดข้อมูลการทำงานจะหยุดจากการทำงานวนรอบ พิมพ์ผลที่เก็บในตัวแปร Max และ Min ทางจอภาพ

แบบฝึกหัดท้ายบทที่ 5 ข้อ 2

จงเขียนโปรแกรมอ่านค่าคะแนนวิชา IT 257 ของนักศึกษา 50 คนทางแป้นพิมพ์ และจงหา

1. คะแนนเฉลี่ยของห้อง
2. นับคะแนนผู้ที่ได้เกรดต่าง ๆ ว่าได้จำนวนอย่างละเท่าใด แบ่งช่วงเกรด และคะแนนดังต่อไปนี้

คะแนนที่สอบได้	เกรดที่ได้
90 – 100	A
76 – 89	B
50 – 75	C
36 – 49	D
0 - 35	F

```

Program test2;
Var    SUM, Score, Count : integer;
        Averscore : real;
        A, B, C, D, F : integer;
Begin
    A := 0; B := 0; C := 0; D := 0; F := 0;           .....(1)
    SUM := 0; Count := 1;                             .....(2)
    While Count <= 50 do                               .....(3)
        begin
            readln (score);                             .....(4)
            case score of                               .....(5)
                90..100 : A := A+1;
                76 .. 89 : B := B+1;
                50 .. 75 : C := C+1;
                36 .. 49 : D := D+1;
                0 .. 35 : F := F+1
            end;
            Sum := Sum + score;                           .....(6)
        end;
        Averscore := Sum/50;                             .....(7)
        writeln ('คะแนนเฉลี่ยมีค่าเท่ากับ ', Averscore); .....(8)
        writeln ('นักศึกษาได้เกรด A=', A,'คน');
        writeln ('นักศึกษาได้เกรด B=', B,'คน');
        writeln ('นักศึกษาได้เกรด C=', C,'คน');
        writeln ('นักศึกษาได้เกรด D=', D,'คน');
        writeln ('นักศึกษาได้เกรด F=', F,'คน');
End.

```


คำอธิบายโปรแกรม

- (1) กำหนดค่าของตัวแปรที่ใช้ในการนับจำนวนผู้ที่ได้เกรดต่างๆ เริ่มต้นให้มีค่าเท่ากับ 0
- (2) กำหนดค่าเริ่มต้นของตัวแปร Sum มีค่าเท่ากับ 0 โดยตัวแปรนี้จะทำหน้าที่เก็บผลบวกของคะแนนของนักศึกษาทั้งหมด และกำหนดให้ตัวแปร Count ให้มีค่าเริ่มต้นเท่ากับ 1 เนื่องจากว่าเราจะใช้ตัวแปรนี้ควบคุมการทำงานวนรอบ
- (3) เป็นคำสั่งการทำงานวนรอบโดยจะวนรอบทั้งหมด 50 รอบ
- (4) อ่านคะแนนของนักศึกษาแต่ละคนในการทำงานแต่ละรอบ
- (5) นำคะแนนของนักศึกษามาเปรียบเทียบตามค่าของคะแนน กรณีอยู่ในช่วงใดจะทำการนับจำนวนเพิ่มขึ้นจากเดิม 1
- (6) นำคะแนนของนักศึกษาเก็บสะสมในตัวแปร Sum
- (7) เมื่อทำงานครบทั้งหมด 50 รอบ จะหลุดจากการทำงานวนรอบพิมพ์ผลลัพธ์ตามที่โจทย์ต้องการทางจอภาพ

แบบฝึกหัดท้ายบทที่ 5 ข้อ 3

จงเขียนเมนูในการให้ผู้ใช้เลือกการทำงานโดยโปรแกรมสามารถหา

1. พื้นที่ของสามเหลี่ยมหน้าจั่ว
2. พื้นที่ของวงกลม
3. พื้นที่ของสี่เหลี่ยม
4. พื้นที่ของทรงกลม

โดยสามารถทำงานได้หลาย ๆ รอบตามที่ต้องการ

```
Program test5;
```

```
Const Pi = 3.14159;
```

```
Var B, H, L : integer;
```

```
R, Area: real;
```

```
ch : char;
```

BEGIN

```
REPEAT .....(1)
    writeln ('   เมนูการทำงาน '); .....(2)
    writeln (' 1. หาพื้นที่ของสามเหลี่ยมหน้าจั่ว');
    writeln (' 2. หาพื้นที่ของวงกลม' );
    writeln (' 3. หาพื้นที่ของสี่เหลี่ยม ');
    writeln (' 4. หาพื้นที่ของทรงกลม' );
    writeln (' 5. จบการทำงานของโปรแกรม');
    write('โปรดเลือกการทำงานที่ท่านต้องการ?'); readln (ch); .....(3)
CASE ch of .....(4)
    '1': begin .....(5)
        write('กรุณาป้อนความยาวฐานและความยาวสูง');
        readln (B, H);
        area := 1/2 *B*H;
        writeln ('พื้นที่ของสามเหลี่ยมหน้าจั่ว =', area:5:2);
        readln;
    end;
    '2': begin .....(6)
        write('กรุณาป้อนรัศมีของวงกลม');
        readln (R);
        area := Pi *R*R;
        writeln ('พื้นที่ของวงกลม =', area:5:2);
        readln;
    end;
    '3': begin .....(7)
        write('กรุณาป้อนความกว้างและความยาว');
```

```

        readln (H,L);
        area := H*L;
        writeln ('พื้นที่ของสี่เหลี่ยม =', area:5:2);
        readln;

    end;
'4': begin
        .....(8)
        write('กรุณาป้อนรัศมีของวงกลม');
        readln (R);
        area := 4/3 *Pi*R*R*R;
        writeln ('พื้นที่ของทรงกลม =', area:5:2);
        readln;

    end;
UNTIL ch='5' ; .
        .....(9)
END.

```

คำอธิบายของโปรแกรม

- (1) คำสั่งการทำงานวนรอบ โดยคำสั่งนี้จะกระทำงานตามคำสั่งและตรวจสอบเงื่อนไขที่ (9)
- (2) เป็นคำสั่งพิมพ์หน้าจอโดยจะพิมพ์รายละเอียดหรือเมนูการทำงาน
- (3) เป็นคำสั่งที่รับการเลือกการทำงานของผู้ใช้เก็บในตัวแปร ch
- (3) เป็นคำสั่งเลือกการทำงานกรณีที่ผู้ใช้ป้อน '1' จะทำงานที่ (5) ถ้าผู้ใช้ป้อน '2' จะทำงานที่ (6) ถ้าผู้ใช้ป้อน '3' จะทำงานที่ (7) ถ้าผู้ใช้ป้อน "4" จะทำงานที่ (8)
- (5) จะทำงานเพื่อรับค่าทางแป้นพิมพ์หาพื้นที่ของสามเหลี่ยม
- (6) จะรับข้อมูลเพื่อคำนวณหาพื้นที่ของวงกลม
- (7) จะรับข้อมูลเพื่อคำนวณหาพื้นที่ของสี่เหลี่ยม
- (8) จะรับข้อมูลเพื่อคำนวณหาพื้นที่ของทรงกลม
- (9) ตรวจสอบว่าตัวแปร ch มีค่าเท่ากับ '5' หรือไม่กรณีถ้าใช่ จะหยุดการทำงาน แต่ถ้าไม่ใช่จะกระทำงานซ้ำไปที่ (1)

แบบฝึกหัดท้ายบทที่ 6

แบบฝึกหัดท้ายบทที่ 6 ข้อ 1

รับข้อความหนึ่งบรรทัดเป็นชื่อที่อยู่ของนักศึกษาคนหนึ่งให้นำข้อมูลนั้นมาพิมพ์ออกในรูปแบบจําหน้าของจดหมาย โดยเมื่อพบเครื่องหมาย / หมายถึงขึ้นบรรทัดใหม่

```
Program test6;
Var
    S : String ;
    L, I : integer;
begin
    write('กรุณาพิมพ์ข้อความ'); readln(S); .....(1)
    L := length(S) ; .....(2)
    For I := 1 to L do .....(3)
    Begin
        If S[I] <> '/' then write(S[I]) .....(4)
        Else writeln;
    End;
End.
```

คำอธิบายโปรแกรม

- (1) พิมพ์ชื่อ ที่อยู่ ความยาวไม่เกิน 256 ตัวอักษร เช่น Urai Lee / 704 moo 8 bangkok
- (2) หาความยาวของข้อความโดยใช้ฟังก์ชัน length
- (3) ทำงานวนรอบจำนวน L ครั้ง
- (4) เปรียบเทียบค่ากรณีที่ไม่ใช่ / จะพิมพ์อักขระนั้นทางจอภาพ กรณีที่ใช่จะขึ้นบรรทัดใหม่

แบบฝึกหัดท้ายบทที่ 6 ข้อ 2

รับข้อความภาษาอังกฤษหนึ่งประโยค จบประโยคด้วยเครื่องหมายจุดทศนิยม จงเขียนโปรแกรม
ภาษาปาสคาลนับจำนวนคำของประโยคนั้น

```
Program test7;
```

```
Var
```

```
    count , word  : integer;
```

```
    ch           : char;
```

```
begin
```

```
    count := 0; word := 0;           .....(1)
```

```
    read (ch);                       .....(2)
```

```
    while ch <> '.' Do                .....(3)
```

```
        begin
```

```
            if ch <> '' then          .....(4)
```

```
                begin
```

```
                    count := count +1;
```

```
                    if count=1 then word := word + 1;
```

```
                end
```

```
            else count := 0;          .....(5)
```

```
            read (ch);                .....(6)
```

```
        end;
```

```
        writeln ('จำนวนคำในประโยคนี้ =', word); .....(7)
```

```
End.
```

คำอธิบายโปรแกรมนี้

ถ้าเมื่อ RUN โปรแกรมและผู้ใช้ป้อนข้อความดังนี้

I am a beautiful girl.

- (1) กำหนดให้ตัวแปร Count และตัวแปร Word มีค่าเท่ากับ 0
- (2) อ่านอักขระตัวแรกของข้อความเก็บในตัวแปร ch
- (3) ทำงานวนรอบในกรณีที่ไม่ใช่ข้อมูลตัวสุดท้ายในที่นี้คือจุดทศนิยม ถ้าเป็นจุดทศนิยมจะไปทำงานที่ (7)
- (4) กรณีถ้าอักขระที่อ่านไม่ใช่ช่องว่างแสดงว่าเป็นตัวอักษรใดๆให้นับจำนวนอักขระโดยเพิ่มค่าของตัวแปร Count ในที่นี้ตัวแปร Count จะนับอักขระในคำๆหนึ่ง และถ้า Count มีค่าเท่ากับ 1 แสดงว่าเป็นอักขระตัวแรกของคำ ให้เพิ่มค่าของ Word ขึ้น 1 ตัวแปร Word จะนับจำนวนของคำในประโยคนี้
- (5) ในกรณีที่อักขระที่อ่านเป็นช่องว่าง แสดงว่าหมดคำเพราะช่องว่างจะอยู่ระหว่างคำ ให้เคลียร์ค่าของ Count ให้มีค่าเท่ากับ 0 เพื่อหาคำใหม่ต่อไป
- (6) อ่านอักขระตัวถัดไป แล้วนำไปตรวจสอบเงื่อนไขที่ (3)
- (7) กรณีที่พบจุดทศนิยม จะหยุดการทำงานวนรอบและพิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 6 ข้อ 3

รับข้อมูลเป็นนิพจน์ทางคณิตศาสตร์ทางแป้นพิมพ์ไทยใช้เครื่องหมาย + - * / จบลงด้วยเครื่องหมาย = ให้คำนวณผลลัพธ์ของนิพจน์นั้น โดยคำนวณตามเครื่องหมายเรียงจากซ้ายไปขวา โดยไม่ต้องคำนึงถึงลำดับความสำคัญของเครื่องหมาย

```
Program test8;
```

```
VAR Ch , Sign : Char ;
```

```
N : Integer;
```

```
T : Real;
```

```
BEGIN
```

```
Write('กรรณำป้อนข้อความที่เป็นนิพจน์ทางคณิตศาสตร์ใดๆ');
```

```

T := 0.0 ; Sign := '+' ; .....(1)
While Sign <> '=' do .....(2)
Begin
    N := 0 ; .....(3)
    Read(ch); .....(4)
    While (Ch<='0') and (Ch>='9') do .....(5)
    Begin
        N := 10 *N + Ord(Ch) – Ord('0') ; .....(6)
        Read(Ch); .....(7)
    End;
    Case Ch of .....(8)
        '+' : T := T + N;
        '-' : T := T - N;
        '*' : T := T * N;
        '/' : T := T / N ;
    End;
    Sign := Ch ; .....(9)
End;
Write('ผลลัพธ์ของนิพจน์นี้มีค่าเท่ากับ' , T : 10:2); .....(10)

```

END.

คำอธิบายโปรแกรม

เมื่อนำโปรแกรมนี้ไปทำงานโดยป้อนนิพจน์ดังนี้ $32*5+7=$ การทำงานของโปรแกรม

- (1) กำหนดค่าเริ่มต้นให้แก่ตัวแปร T ให้มีค่าเท่ากับ 0.0 โดยหน้าที่ในโปรแกรมนี้จะทำหน้าที่เก็บผลลัพธ์ของการคำนวณของนิพจน์โดยกระทำจากซ้ายไปขวา และ Sign เป็นตัวแปรสำหรับเก็บเครื่องหมายที่ใช้ในการคำนวณ กำหนดเริ่มต้นให้มีค่าเท่ากับ '+'
- (2) เป็นการทำงานวนรอบในกรณีที่ไม่ใช่ตัวสุดท้ายในที่นี้คือเครื่องหมายเท่ากับจะกระทำ (3) ถึง (9) กรณีที่ใช่จะกระทำที่ (10)

- (3) การทำงานภายใน loop กำหนดค่าเริ่มต้นให้แก่ตัวแปร N ซึ่งในที่นี้เป็นตัวแปรที่เก็บค่าที่ได้จากการแปลงอักขระที่เป็นตัวเลขให้เป็นค่าที่ใช้ในการคำนวณได้
- (4) อ่านอักขระจากนิพจน์ที่ป้อนเข้ามา 1 ตัวอักษร
- (5) ทำงานวนรอบในกรณีที่ยังอักขระเป็นตัวเลข กรณีที่เป็นตัวกระทำไปที่ (8)
- (6) แปลงอักขระให้เป็นตัวเลขเพื่อใช้ในการคำนวณ
- (7) อ่านอักขระตัวต่อไปต่อนั้น ไป (5)
- (8) ในกรณีที่อักขระที่อ่านไม่ใช่ตัวเลข แสดงว่าเป็นตัวกระทำตัวใดตัวหนึ่ง ซึ่งเก็บในตัวแปร Ch การทำงานของคำสั่งนี้ จะนำค่าของข้อมูลที่เป็นอักขระแต่แปลงเป็นค่าที่เก็บในตัวแปร N กระทำตามตัวกระทำ(operator) โดยเก็บผลจากการคำนวณในตัวแปร T ตัวอย่าง ถ้าผู้ใช้ป้อน 3-4= การทำงานของโปรแกรม จะแปลงอักขระ '3' ให้เป็นค่า 3 เก็บในตัวแปร N ต่อกจากนั้นจึงนำค่า 3 เก็บในตัวแปร T เนื่องจากกำหนดค่า Sign เริ่มต้นให้มีค่าเป็นบวก ในที่นี้ Ch เก็บอักขระ '-' เอาไว้
- (9) นำอักขระที่เป็นตัวกระทำเก็บในตัวแปร Sign แทนตัวเก่าเพื่อใช้ในการคำนวณรอบต่อไป ต่อกจากนั้นจะกระทำงานวนรอบไปที่ (2) กระทำตามลำดับจะเห็นว่า จะทำการแปลงอักขระ '4' ให้เป็นค่าต่อกจากนั้น Ch เก็บเครื่องหมาย '=' จะหลุดจากการทำงานวนรอบ คำนวณค่า จากตัว Sign ในที่นี้คือเครื่องหมายลบที่เก็บไว้ตั้งแต่รอบที่แล้ว ได้ผลลัพธ์เก็บในตัวแปร T และนำ เครื่องหมายเท่ากับให้ค่าแก่ตัวแปร Sign มีผลให้เงื่อนไขเป็นเท็จ
- (10) พิมพ์ผลของการทำงานทางจอภาพกำหนดทศนิยม 2 ตำแหน่ง

แบบฝึกหัดท้ายบทที่ 7

แบบฝึกหัดท้ายบทที่ 7 ข้อ 1

จงรับค่าเดือนทางแป้นพิมพ์จงหาว่าเดือนที่แล้ว และเดือนถัดไปคือเดือนอะไร มีกี่วัน


```

program      ex1;
var    m , lm , sm:1..12;      {lm หมายถึงเดือนที่แล้ว sm หมายถึงเดือนถัดไป}
      dlm, dsm  :28..31;      {dlm คือจำนวนวันของเดือนที่แล้ว }
begin
      {dsm คือจำนวนวันของเดือนถัดไป}
      write ('เดือนที่ท่านต้องการตรวจสอบ ='); readln(m);      .....(1)
      if m=1 then lm=12      .....(2)
      else lm:=pred(m);
      if m=12 then sm:=1      .....(3)
      else sm := succ (m);
      case lm of      .....(4)
          1,3,5,7,8,10,12 : dlm:=31
          2      : dlm:=28;
          4,6 ,9,11 : dlm:=30
      end;
      case sm of      .....(5)
          1,3,5,7,8,10,12 : dlm:=31
          2      : dlm:=28;
          4,6,9,11      : dlm:=30
      end;
      writeln ('เดือนปัจจุบันคือ=', m)      .....(6)
      writeln ('เดือนหลังคือ=' , lm , 'มี=',dlm,'วัน');
      writeln ('เดือนถัดไปคือ=' , sm,'มี=',slm,'วัน');
end.

```

คำอธิบายโปรแกรม

- (1) รับเดือนที่ต้องการทางแป้นพิมพ์โดยป้อนในช่วง 1 ถึง 12
- (2) ถ้าเดือนที่รับมาเป็นเดือนมกราคม เดือนที่แล้วคือเดือนธันวาคม แต่ถ้าไม่ใช่คือลดค่าของเดือนลงหนึ่งลำดับ

- (3) ถ้าเดือนที่รับมาเป็นเดือนธันวาคม เดือนถัดไปคือเดือนมกราคม แต่ถ้าไม่ใช่ให้เพิ่มลำดับของเดือนขึ้นหนึ่งลำดับ
- (4) หาจำนวนวันในเดือนที่แล้ว กรณีที่ลงท้ายด้วยคม แสดงว่าเดือนที่แล้วมี 31 วันกรณีที่ เป็นเดือนกุมภาพันธ์ จำนวนวันเท่ากับ 29 วัน และกรณีที่ เป็นเดือนที่ลงท้ายด้วยยน จำนวนวันมีเท่ากับ 30 วัน
- (5) ทำนองเดียวกับข้อ(4) แต่หาจำนวนวันของเดือนถัดไปแทน
- (6) พิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 7 ข้อ 2

จงรับอักขระทางแป้นพิมพ์ 1 บรรทัด จงหาว่ามีตัวอักษรภาษาอังกฤษ ตัวเลข และช่องว่างอย่าง ละกี่ตัว โดยใช้คำสั่ง CASE..OF

```

program      ex2;
var         ch : char;
           n, c , blank : integer;
begin
    n:=0 ; c:=0 ; blank:=0 ;           .....(1)
    read(ch) ;                          .....(2)
    while ch <> "*" do                    .....(3)
        begin
            case ch of                    .....(4)
                'A'..'Z' , 'a'..'z' : c:=c+1;
                '0'..'9'           : n:=n+1
                "                   : blank:= blank+1;
            end;
            read(ch);                      .....(5)
        end;
    end;

```

```
writeln('จำนวนตัวอักษรที่เป็นตัวเลขมีทั้งหมด=',c); .....(6)
writeln('จำนวนตัวอักษรที่เป็นตัวอักษรภาษาอังกฤษมีทั้งหมด=',n);
writeln('จำนวนตัวอักษรที่เป็นช่องว่าง=',blank);
```

end.

คำอธิบายการทำงานของโปรแกรม

- (1) กำหนดตัวแปรนับอักษรที่เป็นตัวเลข ตัวอักษรภาษาอังกฤษ และช่องว่างให้มีค่า= 0
- (2) อ่านอักขระตัวแรกจากข้อความ การป้อนสมมติว่าป้อนดังนี้
IT257H Pascal Programming* สำหรับโปรแกรมนี้อ่านอักขระที่เครื่องหมาย *
- (3) การทำงานใช้ While .. do เพื่อตรวจสอบการทำงานกรณีที่ไม่ใช่ * ซึ่งเป็นข้อมูลตัวสุดท้ายจะทำงานวนรอบ แต่ถ้าเป็นข้อมูลตัวสุดท้ายจะหลุดจากการทำงานวนรอบ
- (4) นำอักขระมาตรวจสอบโดยใช้คำสั่ง Case..of กรณีที่เป็นตัวอักษรภาษาอังกฤษตัวเล็กหรือตัวใหญ่ก็ตามจะเพิ่มค่าตัวนับ C ขึ้น 1 แต่ถ้าเป็นอักษรที่เป็นตัวเลขจะเพิ่มค่าตัวนับ N ขึ้น 1 แต่ถ้าเป็นอักขระที่เป็นช่องว่างจะเพิ่มค่าตัวนับ blank ขึ้น 1 กรณีอื่นๆจะไม่มีการทำงานอะไร
- (5) อ่านอักขระตัวถัดไป ต่อจากนั้นวนรอบไปทำ(3)
- (6) เป็นคำสั่งในการพิมพ์ผลลัพธ์จากการทำงานเมื่อสิ้นสุดการทำงานวนรอบ

แบบฝึกหัดท้ายบทที่ 7 ข้อ 3

การลงคะแนนเลือกหัวหน้ากลุ่มของสมาชิกจำนวน 50 คน มีผู้ลงสมัคร 4 คนคือหมายเลข 1 , 2 , 3 และ 4 ตามลำดับ จงเขียนโปรแกรมนับจำนวนคะแนนเสียงของทั้งสี่หมายเลขและหาว่าหมายเลขใดชนะการเลือกตั้งครั้งนี้

```
program ex3;
var A : array[1..4] of integer; .....(1)
    l , max , number : integer;
Begin
    For l=1 to 4 do A[l]:=0; .....(2)
```

```

For I:=1 to 50 do .....(3)
    Begin
        Read (number); .....(4)
        A[number] := A[number+1]; .....(5)
    End;
Max:=1 ; .....(6)
For I := 2 to 4 do .....(7)
    Begin
        If A[I] > A[Max] then .....(8)
            Max := I
        End;
For I := 1 to 4 do .....(9)
    Writeln('หมายเลข ' , I , ' ได้ทั้งหมด ' , A[I] , ' คะแนน');
Writeln ('ผู้ที่ได้คะแนนมากที่สุดได้แก่หมายเลข ' , Max); .....(10)
End.

```

คำอธิบายโปรแกรม

- (1) ประกาศให้ตัวแปร A เป็นตัวแปรชนิดแถวมี ตัวชี้ตั้งแต่ 1 ถึง 4 ซึ่งตัวชี้นี้แทนหมายเลขของผู้สมัครนั่นเอง ข้อมูลที่เก็บอยู่เป็นจำนวนนับคะแนนของผู้เลือกในแต่ละหมายเลข
- (2) กำหนดค่าเริ่มต้นของคะแนนทุกหมายเลขให้มีค่าเท่ากับ 0
- (3) ทำงานวนรอบ 50 รอบ (4)-(5)
- (4) ในแต่ละรอบจะอ่านหมายเลขของการเลือกหัวหน้ากลุ่มของแต่ละคน เก็บในตัวแปร number
- (5) ถ้า number ตรงกับเลขใด จะเพิ่มตัวนับของหมายเลขนั้นของตัวแปร A ขึ้น 1
- (6) เมื่อทำงานครบ 50 รอบ จะหาว่าใครได้คะแนนมากที่สุด โดยกำหนดให้หมายเลข 1 เป็นหมายเลขที่มีคะแนนมากที่สุดก่อน
- (7) นำคะแนนของแต่ละหมายเลขมาเปรียบเทียบกันเพื่อหาว่าใครได้คะแนนมากที่สุด โดยจะทำการเปรียบเทียบกับหมายเลขที่ 2 หมายเลขที่ 3 และหมายเลขที่ 4 ตามลำดับ

- (8) ถ้าหมายเลขใดได้คะแนนมากกว่าในการเปรียบเทียบกัน จะนำมาเก็บในตัวแปร Max
- (9) พิมพ์ผลของการเลือกตั้งของทุกหมายเลข
- (10) พิมพ์สรุปว่าใครได้คะแนนมากที่สุด

แบบฝึกหัดท้ายบทที่ 8

แบบฝึกหัดท้ายบทที่ 8 ข้อ 1

จงเขียนโปรแกรมคูณเมตริกซ์

```

Program      Test1;
Const       N = 3 ;
Var        A, B, C : array [1..N , 1..N] of integer;          .....(1)
           I , J , K , SUM : integer;
Begin
  For I := 1 to N do                                          .....(2)
    For J := 1 to N do
      Readln(A[I, J]);
  For I := 1 to N do                                          .....(3)
    For J := 1 to N do
      Readln(B[I, J]);
  For I := 1 to N do                                          .....(4)
    For J := 1 to n do
      Begin
        SUM := 0;
        For K := 1 to N do                                     .....(5)
          SUM := SUM + A[I,K] * B[K,J] ;

```

```

                                C[I,J] := SUM ;                .....(6)
                                End;
                                For I := 1 to N do                .....(7)
                                    Begin
                                        For J := 1 to N do
                                            Write(C[I,J] : 5);
                                            Writeln;
                                        End;
                                    End;
                                END.

```

คำอธิบายโปรแกรม

- (1) กำหนดตัวแปร A,B,C เป็นตัวแปรอาเรย์ 2 มิติขนาด 3x3
- (2) รับข้อมูลเป็นเลขจำนวนเต็ม 9 ตัวเก็บในตัวแปรอาเรย์ A
- (3) รับข้อมูลเป็นเลขจำนวนเต็ม 9 ตัวเก็บในตัวแปรอาเรย์ B
- (4) ทำงานวนรอบเป็น loop ซ้อน loop เพื่อหาผลคูณของเมตริกซ์ AxB
- (5) ทำการคำนวณค่าของผลคูณโดยเก็บสะสมในตัวแปร Sum
- (6) ผลจากการคูณนำมาเก็บในตัวแปร C ตำแหน่งที่ I , J
- (7) พิมพ์เมตริกซ์ C ซึ่งเก็บผลลัพธ์จากการคูณพิมพ์ออกมาทางจอภาพ

แบบฝึกหัดท้ายบทที่ 8 ข้อ 2

จงรับอักขระทางแป้นพิมพ์จำนวนหนึ่ง จงพิมพ์อักขระจากตัวสุดท้ายย้อนกลับถึงตัวแรก

```

Program      test2;
var      A : array[1..100] of char;
          N , I : integer;
          ch : char;

Begin
          N := 0 ;                .....(1)

```

```

read(ch); .....(2)
While ch <>'*' do .....(3)
    Begin
        N := N + 1; .....(4)
        A [N]: =ch; .....(5)
        read (ch); .....(6)
    End;
For I := N down to 1 do .....(7)
    Write (A [I]);

```

End.

คำอธิบายโปรแกรม

- (1) กำหนดตัวแปร N เป็นตัวแปรในการนับจำนวนตัวอักษรที่ป้อนมีค่าเริ่มต้น=0
- (2) อ่านอักขระตัวแรกทางแป้นพิมพ์ ในการป้อนสามารถป้อนเป็นข้อความได้ไม่เกิน 100 ตัวอักษร โดยตัวสุดท้ายให้ป้อน * เช่น urai*
- (3) ทำงานวนรอบจนกระทั่งตัวอักษรที่อ่านเป็น *
- (4) เพิ่มตัวนับข้อมูลขึ้น 1
- (5) นำตัวอักษรที่อ่านเข้ามาเก็บในตัวแปรอาเรย์ A ตัวที่ N
- (6) อ่านอักขระตัวถัดไป
- (7) เมื่อออกจากการทำงานวนรอบ พิมพ์ตัวอักษรที่เก็บในอาเรย์ A โดยเริ่มจากตัวสุดท้ายย้อนจนกระทั่งพิมพ์ตัวแปร เป็นอันจบโปรแกรม

แบบฝึกหัดท้ายบทที่ 8 ข้อ 3

จงเขียนโปรแกรมภาษาปาสคาลรับคะแนนของนักศึกษาตั้งแต่เลขที่ 25 ถึง 50 ให้นำว่านักศึกษาเลขที่ใดได้คะแนนสูงสุด และนักศึกษาเลขที่ใดได้คะแนนต่ำสุด

```

Program Test3;
var A : array[25..50] of integer;

```

```

Min, Max , I : integer;
Begin
  For I := 25 to 50 do .....(1)
    Readln(A[I]);
  Max := 25; .....(2)
  Min := 25;
  for I := 26 to 50 do .....(3)
    Begin
      If A[I]>A[Max] then Max := I ; .....(4)
      if A [I]<A[Min] then Min := I;
    End;
  Writeln ('นักศึกษาเลขที่' , Max, 'ได้คะแนนมากที่สุด' ); .....(5)
  Writeln ('นักศึกษาเลขที่' , Min, 'ได้คะแนนน้อยที่สุด' );
End.

```

คำอธิบายโปรแกรม

- (1) อ่านคะแนนของนักศึกษาเลขที่ 25 ถึง 50 เก็บในตัวแปรอาเรย์ A ตัวชี้ที่ 25 ถึง 50 ตามลำดับ
- (2) กำหนดให้เลขที่ที่ได้คะแนนมากที่สุดและน้อยที่สุด เป็นเลขที่ 25 เป็นการสมมุติเบื้องต้น
- (3) นำค่าที่สมมุติในข้อ(2) มาทำการเปรียบเทียบกับหมายเลข 26 ถึง 50
- (4) การเปรียบเทียบจะเปรียบเทียบค่าที่ละหมายเลขถ้าหมายเลขใดมีคะแนนมากกว่าให้เก็บในตัวแปร Max และถ้าหมายเลขใดมีคะแนนน้อยกว่าให้เก็บในตัวแปร Min
- (5) เมื่อเปรียบเทียบจนครบทุกหมายเลข พิมพ์หมายเลขของนักศึกษาที่ได้คะแนนมากที่สุดและน้อยที่สุด ทางจอภาพ

แบบฝึกหัดท้ายบทที่ 8 ข้อ 4

ให้เขียนโปรแกรมรับค่าความสูงของคนกลุ่มหนึ่ง คำนวณหาค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของข้อมูลกลุ่มนี้ โดยสมมติว่าข้อมูลมีจำนวนไม่เกิน 10 คน

สูตร ส่วนเบี่ยงเบนมาตรฐาน = $\text{SQRT}(\text{SumofSquare}/N - \text{Sqr}(\text{MEAN}))$

MEAN = ค่าเฉลี่ย

SumofSquare = ผลรวมของกำลังสองของข้อมูลแต่ละจำนวน

```
Program      Test4;
var          N, Sum, T : integer;
            Mean , STD , Sq : real;
Begin
    N:=0; Sum:=0; Sq:=0          .....(1)
    Read (T);                    .....(2)
    While T > 0 do                .....(3)
        Begin
            Sum := Sum +T;        .....(4)
            Sq  := Sq+ Sqr(T) ;    .....(5)
            N   := N+1;           .....(6)
            Read(T);              .....(7)
        End;
    Mean := Sum / N ;            .....(8)
    STD  := SQRT(SQ / N - SQR(Mean)) ; .....(9)
    Writeln('ค่าเฉลี่ยมีค่าเท่ากับ', Mean:10:4); .....(10)
    Writeln('ส่วนเบี่ยงเบนมาตรฐานมีค่าเท่ากับ',STD:10:4);
End.
```

คำอธิบายโปรแกรม

- (1) กำหนดค่าเริ่มต้นให้กับตัวแปรที่ใช้ในการนับ การเก็บผลรวม ให้มีค่าเท่ากับ 0
- (2) อ่านค่าความสูงของบุคคลแรกเก็บในตัวแปร T
- (3) การทำงานจะทำงานวนรอบในกรณีที่ความสูงมีค่ามากกว่า 0 กระทำ (4)-(7)
- (4) นำค่าความสูงบวกสะสมในตัวแปร Sum
- (5) นำค่าความสูงยกกำลังสองบวกสะสมในตัวแปร Sq
- (6) นับจำนวนของบุคคลเพิ่มขึ้น 1
- (7) อ่านความสูงของบุคคลคนต่อไป
- (8) หาค่าเฉลี่ยของความสูงของบุคคล N คน
- (9) หาส่วนเบี่ยงเบนมาตรฐานจากสูตรที่กำหนดให้มา
- (10) พิมพ์ผลลัพธ์ออกทางจอภาพกำหนดจุดทศนิยม 4 ตำแหน่ง

แบบฝึกหัดท้ายบทที่ 8 ข้อ 5

รูปสามเหลี่ยมปาสคาลแสดงได้ดังรูปข้างล่างนี้

			1						
			1	1					
			1	2	1				
			1	3	3	1			
			1	4	6	4	1		
			1	5	10	10	5	1	
			1	6	15	20	15	6	1

ซึ่งสามเหลี่ยมที่ได้นี้ก็คือ สัมประสิทธิ์ทวินาม (Binomial Coefficients) นั่นเอง ให้เขียนโปรแกรม เพื่อแสดงสิบแถวแรกของสามเหลี่ยมปาสคาลนี้

```

PROGRAM EX5;
VAR A : Array[0..10] of integer;
    I, J : integer;
BEGIN
    A[0] := 1 ;                               .....(1)
    For I := 1 to 10 do                       .....(2)
        Begin
            A[I] := A[I-1];                   .....(3)
            For J := I-1 downto 2 do          .....(4)
                A[J] := A[J] + A[J-1];      .....(5)
            Gotoxy(40 - (I - 1) * 4 , 1 + I) ; .....(6)
            For J := 1 to I do               .....(7)
                Write(A[J]);
            End;
        End;
    END.

```

คำอธิบายโปรแกรม

- (1) กำหนดตัวแปร A[0] มีค่าเริ่มต้นมีค่าเท่ากับ 1 สำหรับค่าของข้อมูลจะเก็บในช่องที่ 1 ถึงช่องที่ 10 ของตัวแปร A
- (2) ทำงานวนรอบ 10 รอบเพื่อแสดงสิบแถวแรกของสามเหลี่ยมปาสคาล
- (3) กำหนดให้ข้อมูลตำแหน่งสุดท้ายของแต่ละแถวมีค่าเท่ากับข้อมูลตัวสุดท้ายของแถวที่แล้ว ซึ่งในที่นี้คือค่า 1 นั่นเอง
- (4) ทำงานวนรอบเริ่มจากตำแหน่งรองสุดท้ายถัดมาจนถึงช่องที่ 2 สังเกตได้ว่าช่องแรกกับช่องสุดท้ายเป็น 1 ตลอด สำหรับช่องแตกต่างจากนี้ค่าของข้อมูลจะเปลี่ยนตลอด
- (5) ทำการบวกเพื่อหาค่าของข้อมูลอยู่ในตำแหน่งที่ถูกต้อง ตำแหน่งปัจจุบันมีค่าเท่ากับค่าข้อมูลเก่าบวกด้วยค่าของข้อมูลตำแหน่งก่อนหน้า
- (6) หาดำแหน่งในการพิมพ์บนจอภาพ ผู้ใช้สามารถกำหนดได้ให้สวยงาม

- (7) พิมพ์ค่าข้อมูลออกทางจอภาพ ทีละแถว ซึ่งการทำงานจะพิมพ์ทั้งหมด 10 บรรทัดเมื่อครบแล้วเป็นอันจบโปรแกรม

แบบฝึกหัดท้ายบทที่ 8 ข้อ 6

จงเขียนโปรแกรมเพื่ออ่านและพิมพ์ตารางคะแนนของนักศึกษาพร้อมทั้งคำนวณหาคะแนนรวมของนักศึกษาแต่ละคน คะแนนเฉลี่ยสำหรับแต่ละวิชา และคะแนนเฉลี่ยของคะแนนรวมทั้งหมด ดังรูปแบบต่อไปนี้

	Paper1	Paper2	Paper3	Total
Student1	xx	xx	xx	xx
Student2	xx	xx	xx	xx
...
Student N	xx	xx	xx	xx
Average	xx.xx	xx.xx	xx.xx	xx.xx

```

Program      Test6;
Const       N =100;
Var  A      : array[1..N, 1..4] of integer;      ....(1)
          Sum , Average: array[1..4] of real;    ....(2)
          I , J ,K : integer;
          Ch : Char;
Begin
          Ch := 'y' ; I:= 0;                      ....(3)
          Repeat                                  ....(4)
              I := I+1 ;                          ....(5)
              A[I,4] := 0;                        ....(6)
          Until
  
```

```

For J := 1 to 3 do          .....(7)
    begin
        Readln(A[I,J]);    .....(8)
        A[I,4] := A[I,4] + A[I,J] ;    .....(9)
    End;
Write('มีนักศึกษาคนอื่นอีกใช่หรือไม่ ถ้ามี กด Y ถ้าไม่มีแล้วให้กด N');
Read(Ch);                  .....(10)
Until (Ch='n' ) or (Ch='N') ;    .....(11)

For J := 1 to 4 do        .....(12)
    Sum[J] := 0.0 ;

For K := 1 to I do        .....(13)
    For J := 1 to 4 do
        Sum[J] := Sum[J] + A[K,J];

For J := 1 to 4 do        .....(14)
    Average[J] := Sum[J] / I ;

WriteLn('          Paper1  Paper2  Paper3  Total');
For K := 1 to I do        .....(15)
    Begin
        Write('Student', K);
        For J := 1 to 4 do
            Write(A[K,J]:10);
        WriteLn;
    End;

```

Write('Average ');

For J := 1 to 4 do(16)

Write(Average[J]:10:2);

WriteLn;

End.

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปรอาเรย์สองมิติ A ขนาด 100x4 เพื่อเก็บคะแนนของนักศึกษา
- (2) ประกาศตัวแปร Sum เป็นอาเรย์ขนาด 1 มิติเพื่อเก็บผลรวมของคะแนนแต่ละ Paper โดยเก็บในตัวชี้ที่ 1, 2 และ 3 ตามลำดับ ส่วนช่องสุดท้ายเก็บผลรวมทั้งหมด ส่วนตัวแปรอาเรย์ Average ใช้สำหรับเก็บค่าเฉลี่ยของคะแนนแต่ละ Paper สำหรับช่องสุดท้ายเก็บคะแนนเฉลี่ยทั้งหมด
- (3) กำหนดค่าเริ่มต้นให้กับตัวแปร Ch และตัวแปร I ซึ่งในการทำงานกำหนดให้มีนักศึกษามีสูงสุดได้ 100 คน ในที่นี้ไม่ทราบจำนวนนักศึกษา จึงกำหนดตัวแปร I ให้นับจำนวนนักศึกษาทั้งหมด เริ่มต้นให้มีค่าเป็น 0 ก่อน เมื่ออ่านนักศึกษาแต่ละคนเพื่อทำงาน ค่าของตัวแปรจะเพิ่มขึ้นครั้งละ 1 สำหรับตัวแปร Ch เป็นตัวแปรที่กำหนดขึ้นมาเพื่อโต้ตอบกับผู้ใช้โปรแกรม กำหนดให้มีค่าเริ่มต้นให้เท่ากับอักขระ y
- (4) เป็นคำสั่งเริ่มต้นของการทำงานวนรอบ ซึ่งคู่กับคำสั่งที่ (11)
- (5) เมื่อเริ่มทำงานจะรับข้อมูลนักศึกษา จะเพิ่มตัวนับจำนวนนักศึกษา I ให้เพิ่มขึ้น 1
- (6) กำหนดให้ผลรวมของคะแนนของคนที่ I มีค่าเริ่มต้นเท่ากับ 0
- (7) เป็นการทำงานวนรอบ 3 รอบเพื่อรับคะแนนของ Paper ทั้งสาม และหาผลรวม
- (8) อ่านคะแนนของ Paper1, Paper2, Paper3 เก็บในตัวแปร A[I,1], A[I,2], A[I,3] ตามลำดับ โดยที่ I คือนักศึกษาคนที่ I นั่นเอง
- (9) สำหรับ A[I,4] จะเก็บผลรวมของทั้งสาม Paper ที่ป้อนเข้ามา
- (10) เมื่อรับคะแนนของคนที่ I เรียบร้อยแล้ว จะถามผู้ใช้ว่าจะป้อนคะแนนของนักศึกษาคนต่อไปหรือไม่ ผู้ใช้ต้องป้อนข้อมูลเพื่อโต้ตอบ โดยข้อมูลที่ผู้ใช้ป้อนเก็บในตัวแปร Ch

- (11) เป็นเงื่อนไขในการทำงานวนรอบ ถ้าผู้ใช้ไม่ต้องการทำงานกด N โปรแกรมจะหยุดการทำงานวนรอบ แต่ถ้ามีนักศึกษาคนอื่น ๆ อีก จะกระทำงานวนรอบ (4)-(11) เพื่อนับจำนวนนักศึกษาเพิ่มขึ้นรับคะแนนของ Paper และหาผลรวมของคะแนนทั้งหมด จนกระทั่งผู้ใช้กด N
- (12) เมื่อป้อนข้อมูลครบทุกคนแล้วในที่นี้ ค่า I เก็บจำนวนของนักศึกษาทั้งหมดไว้ เราสามารถหาผลรวมของคะแนนของ Paper แต่ละ Paper ได้ โดยก่อนอื่นต้องกำหนดค่าเริ่มต้นให้กับ ตัวแปร Sum ให้มีค่าเท่ากับ 0 เสียก่อน
- (13) คำสั่งนี้เป็นการบวกสะสมคะแนนของแต่ละ Paper กรณี Paper1 จะเก็บสะสมใน Sum[1] ถ้าเป็น Paper2 เก็บคะแนนสะสมใน Sum[2] Paper ที่สามก็ทำนองเดียวกัน ส่วน Sum[4] จะเก็บผลรวมของคะแนนทั้งหมดแทน
- (14) นำมาหาค่าเฉลี่ย โดยนำค่า Sum ของทุกชองหารด้วย I ซึ่งเป็นจำนวนของนักศึกษาทั้งหมด ผลลัพธ์เป็นค่าเฉลี่ยนั่นเอง
- (15) พิมพ์ผลลัพธ์ที่โจทย์ต้องการโดยพิมพ์คะแนนและผลรวมคะแนนทุก Paper ของนักศึกษาแต่ละคน จำนวนทั้งหมด I คนออกทางจอภาพ
- (16) พิมพ์ค่าเฉลี่ยของคะแนนของทุก Paper และค่าเฉลี่ยทั้งหมด ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 9

แบบฝึกหัดท้ายบทที่ 9 ข้อ 1

จงเขียนฟังก์ชัน ที่สามารถบอกว่าเลขชนิดจำนวนเต็มเป็นตัวประกอบปฐมหรือไม่ (Prime number) ตัวประกอบปฐมนี้คือเลขที่ไม่มีเลขใดหารลงตัวนอกจากตัวเองและ 1

```
FUNCTION PRIME_NUMBER(X:INTEGER) : BOOLEAN ;
```

```
VAR I, J : integer ;
```

```
    P : Boolean;
```

```
    {
```

BEGIN

J := X div 2 ;(1)

P := TRUE;(2)

FOR I := 2 to J DO(3)

 If X mod I = 0 then P := FALSE ;(4)

Prime_Number := P ;(5)

END;

คำอธิบายโปรแกรม

- (1) การหาว่ามีตัวเลขตัวใดหารเลขจำนวนเต็มที่ต้องการลงตัวหรือไม่นั้น โดยทั่วไปตัวเลขที่หารลงตัวนั้นจะอยู่ระหว่าง 1 ถึงตัวเลขนั้น และไม่เกินครึ่งหนึ่งด้วย เราจึงหาข้อมูลที่อยู่ระหว่างกลางเพื่อทำการตรวจสอบก็พอ
- (2) กำหนดตัวแปร P เป็นชนิดบูลีน กำหนดค่าเริ่มต้นให้เป็น TRUE หมายถึงว่าสมมุติว่าตัวเลขที่ต้องการตรวจสอบเป็น Prime number กล่าวคือไม่มีตัวเลขใดหารลงตัว
- (3) กำหนดให้ทำงานวนรอบ ตั้งแต่ 2 ถึงค่า จุดกึ่งกลางของข้อมูลที่ต้องการตรวจสอบ
- (4) ตรวจสอบว่ามีเลขจำนวนเต็มใดที่หารลงตัวบ้าง กรณีที่ไม่มีค่าของ P จะคงเดิมแต่ถ้ามีตัวเลขจำนวนเต็มใดหารลงตัว จะเปลี่ยนค่า P เป็น FALSE แสดงว่าตัวเลขนี้ไม่เป็น Prime number
- (5) นำผลลัพธ์ที่ได้จากการทำงาน ให้ค่าแก่อ็ฟังก์ชัน เพื่อส่งค่านี้กลับไปยังจุดเรียกใช้

แบบฝึกหัดท้ายบทที่ 9 ข้อ 2

ดอกเบี้ยเงินกู้คิดจากผลคูณของเงินต้น ระยะเวลากู้เป็นเดือน และอัตราดอกเบี้ยต่อเดือน โดยจะผ่อนชำระเงินต้นพร้อมดอกเบี้ยเป็นเงินเท่ากันทุกเดือน จงเขียนฟังก์ชัน หาเงินผ่อนชำระรายเดือนโดยให้เงินต้น ระยะเวลากู้ และอัตราดอกเบี้ยเป็นพารามิเตอร์ และใช้ฟังก์ชันนี้เขียนโปรแกรมอ่านอัตราดอกเบี้ย แล้วพิมพ์เป็นตารางแสดง เงินค้างวดรายเดือนสำหรับเงินต้น 100, 200 , 300 , ... ด้วยระยะเวลากู้ 3, 6 , ... 24 เดือน

Program EX9_2;

VAR

I ,J : integer;

Rate : real;

Function Money_Month(M:real ; Month : integer ; rate : real) : real ;(1)

Begin

Money_Month := (M+(rate/100*M*Month)) / Month;

End;

BEGIN

Write('อัตราดอกเบี้ยต่อเดือนคือ'); readln(rate);(2)

Writeln('เงินต้น 100 200 300 400 500 600 700 800 900 1000');

Writeln('เงินกู้ / เดือน');

For I := 1 to 8 do(3)

Begin

Write(3 * I); {ระยะเวลาที่กู้}(4)

For J := 1 to 10 do(5)

Write(Money_Month(J*100, 3*I , rate):5:2);(6)

Writeln;

End;

END.

คำอธิบายโปรแกรม

- (1) เป็นการสร้างฟังก์ชันเพื่อหาเงินผ่อนชำระรายเดือน หลักการคิดคือเทียบบัญชีดอกเบี้ยรายงวด โดยคิดหาอัตราดอกเบี้ยที่ต้องจ่ายทั้งหมดเสียก่อนนำมารวมกับเงินต้น และหารด้วยจำนวนเดือนทั้งหมด
- (2) รับอัตราดอกเบี้ยทางแป้นพิมพ์
- (3) ทำงานวนรอบ 8 รอบ โดยทำงานต่อใน (4)-(6)

- (4) นำค่า I*3 ในการทำงานแต่ละรอบมีผลให้ ในแต่ละรอบจะพิมพ์ 3, 6, 9, 12, 15, 18, 21, 24 ซึ่งคือระยะเวลาที่ที่ต้องการหาค่าเงินงวดนั่นเอง
- (5) ทำงานวนรอบอีก 10 ครั้ง
- (6) เป็นการเรียกใช้ฟังก์ชันโดยส่ง เงินต้น 10 ค่าด้วยกัน ได้แก่ 100,200, ...,1000 ใน การทำงานในแต่ละรอบของ I ดังนั้นการเรียกใช้ฟังก์ชันนี้จะเรียกใช้ถึง 10*8 ครั้งนั่นเอง ผลจากการทำงานจะพิมพ์ผลของการทำงานออกทางจอภาพเป็นตารางของเงินค่างวดรายเดือนสำหรับเงินต้น 100,200, ...,1000 ด้วยระยะเวลาที่ 3, 6, 9, 12, 15, 18, 21, 24 เดือน ซึ่งในที่นี้ใช้อัตราดอกเบี้ยต่อเดือนเท่ากันหมดคือ rate นั่นเอง

แบบฝึกหัดท้ายบทที่ 9 ข้อ 3

จงเขียนโปรแกรมอ่านเลขโรมัน แล้วพิมพ์เลขโรมันนั้นพร้อมทั้งเลขอารบิกที่มีค่าเท่ากัน โดยสร้างฟังก์ชันซึ่งรับเลขโรมันเป็นกลุ่มอักขระขีดซ้าย ให้ผลเป็นเลขอารบิกจำนวนเต็มที่มีค่าเท่ากัน ระวังการผสมกันเช่น IV ต้องแปลงเป็น 4 และต้องตรวจสอบได้ว่าผิดรูปแบบการเขียนเลขโรมันหรือไม่

```
PROGRAM EX9_3;
```

```
VAR   S : String;           {ตัวแปรเก็บเลขโรมันใดๆ}
      Sum,                  {เก็บผลลัพธ์เป็นผลรวมของเลขอารบิก}
      Number ,             {เก็บผลจากการแปลงเลขโรมันจากฟังก์ชัน Change}
      Pre ,                {เก็บค่าของเลขตัวก่อนในกรณีที่มีเลขโรมันหลายตัว}
      L ,                  {ความยาวของเลขโรมัน}
      I : Integer;
```

```
FUNCTION Change(Ch:Char):integer;
```

```
BEGIN
```

```
  Case Ch of
```

```
    'I'   : Change := 1;
```

```
    'V'   : Change := 5;
```

```

'X'   : Change := 10;
'L'   : Change := 50;
'C'   : Change := 100;
'D'   : Change := 500
'M'   : Change := 1000;

```

END;

BEGIN

Write('ป้อนเลขโรมันที่ท่านต้องการแปลงเป็นเลขอารบิก');read(S);

L := Length(S);

Sum := 0; Pre :=1000 ;(1)

For I := 1 to L do(2)

BEGIN

Number := Change(S[I]);(3)

If Pre>=Number then(4)

Sum := Sum+Number

Else

Sum := Number – Sum ;

Pre := Number;(5)

END;

WriteLn('เลขโรมันนี้มีค่าเท่ากับ', Sum);(6)

END.

คำอธิบายโปรแกรม

การทำงานของโปรแกรมนี้สมมุติผู้ใช้ป้อน

XVII ผลของการทำงาน $10+5+1+1 = 17$

LXVI ผลของการทำงาน $50+10+5+1=66$

IV ผลของการทำงาน $5 -1 = 4$

XL ผลของการทำงาน $50-10 = 40$

- (1) รับข้อความเป็นเลขโรมันใดๆ หาความยาวเก็บในตัวแปร L ต่อจากนั้นกำหนดค่าเริ่มต้นให้กับตัวแปร Sum ให้มีค่าเป็นศูนย์ และตัวแปร Pre ให้มีค่าเท่ากับค่าที่มากที่สุดของเลขโรมันเพื่อใช้สำหรับตรวจสอบการผสมกันของเลขโรมัน
- (2) ทำงานวนรอบทั้งหมด L รอบ
- (3) เรียกใช้ฟังก์ชัน Change เพื่อเปลี่ยนเลขโรมันตัวที่ i ให้เป็นเลขอารบิก เก็บในตัวแปรชื่อ Number
- (4) เปรียบเทียบลำดับของตัวเลขโรมันว่าผสมกันอย่างไร ในกรณีที่เรียงจากมากไปน้อยจะนำค่าของตัวเลขบวกสะสมในตัวแปร Sum ถ้าเรียงลำดับจากน้อยไปมาก จะนำมาหักลบกัน เก็บในตัวแปร Sum เช่นกัน
- (5) ต่อจากนั้นนำค่าของข้อมูลตัวปัจจุบันให้แก่ตัวแปร Pre เพื่อใช้ทดสอบค่าในรอบต่อไป การทำงานจะกระทำวนรอบจาก (2)-(5)
- (6) เมื่อหลุดจากการทำงานวนรอบจะพิมพ์ผลลัพธ์ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 9 ข้อ 4

จงเขียนฟังก์ชันเรียกตัวเองหาผลรวมของเลข $1+2+3+\dots+N$

```
FUNCTION SUM(X:INTEGER);
BEGIN
    IF X=0 THEN SUM := 0
    ELSE SUM := X + SUM(X-1) ;
END;
```

คำอธิบาย

การทำงานของฟังก์ชันนี้เพียงแต่ใช้เงื่อนไขในการทำงาน ในกรณีที่ค่า $X > 0$ จะเรียกตัวเองโดยนำค่าตัวเอง บวกกับค่าของผลการทำงานของลำดับที่แล้ว เรียกซ้อนกันไปเรื่อยๆ จนกระทั่ง X มีค่าเท่ากับ 0 ซึ่งจะส่งผล 0 กลับมา และย้อนกลับมาจนกระทั่งได้ผลลัพธ์ที่ต้องการ

แบบฝึกหัดท้ายบทที่ 10

แบบฝึกหัดท้ายบทที่ 10 ข้อ 1

จงเขียนไพธอนในการวนช่อง N ช่อง

```
Procedure space(N:Integer);
```

```
Var I:Integer;
```

```
Begin
```

```
    for I := 1 to N do
```

```
        Write (' ');
```

```
End;
```

คำอธิบาย

การทำงานของไพธอนนี้เป็นการส่งผ่านค่าทางเดียว ซึ่งในที่นี้ตัวแปร N จะรับค่าเพื่อความคุมการทำงานวนรอบนั่นเอง

แบบฝึกหัดท้ายบทที่ 10 ข้อ 2

เขียนโปรแกรมในการสลับค่าสองค่าของข้อมูล

```
Program Swap_Data;
```

```
Var A,B :Integer;
```

```
Procedure SWAP( Var A,B : Integer);
```

```
Var Temp : Integer;
```

```
Begin
```

```
    Temp := A ;
```

```
    A := B ;
```

```
    B := Temp;
```

```
End;
```

Begin { Main Program)

Writeln(' Before swap');

Write('A='); Read(A);

Write('B=');readln(B);

SWAP (A,B);

Writeln(' After swap);

Writeln('A=',A);

Writeln('B=',B);

END.

คำอธิบาย โปรแกรมนี้มีการเรียกใช้โพรซีเจอร์ Swap ซึ่งมีการส่งผ่านค่าแบบอ้างอิงซึ่งผลการทำงานในโพรซีเจอร์จะส่งผลกระทบต่อตัวแปรจากจุดเรียกใช้ที่อ้างอิงกัน มีผลให้ค่าของตัวแปรในโปรแกรมหลักเปลี่ยนแปลง

แบบฝึกหัดท้ายบทที่ 10 ข้อ 3

จงเขียนโปรแกรมรับข้อความทางแป้นพิมพ์ หลายบรรทัด จงจัดข้อมูลต่างๆ เหล่านั้นให้พิมพ์ บรรทัดละ 2 คำ

```
Program Print_Data;
Var   Ch : Char;
      N , Word : Integer;
Begin
    N := 0; Word := 0; .....(1)
    While not EOF Do .....(2)
    Begin
        Read(Ch); .....(3)
        Case Ch of
            ' ' : Begin .....(4)
                    N := 0 ;
                    If Word = 2 then
                        Begin
                            Writeln;
                            Word := 0;
                        End;
                    End;
            Else Begin .....(5)
                    N := N + 1 ;
                    If (N=1) then Word := Word + 1;
                End;
        End;
    End;
    Write(Ch); .....(6)
```

End;

END.

คำอธิบายโปรแกรม

การทำงานของโปรแกรมนี้อาศัยการป้อนข้อความได้หลายบรรทัดสิ้นสุดที่ Ctrl-Z ดังตัวอย่าง

Pascal Programming And Data

Structure And Introduction

To Computer Science

กด Ctrl-Z

- (1) กำหนดให้ตัวแปร N มีค่าเริ่มต้นเท่ากับ 0 ในที่นี้ตัวแปรนี้ทำหน้าที่นับจำนวนของอักษรในคำๆหนึ่ง สำหรับตัวแปร Word เป็นตัวแปรที่นับจำนวนคำในที่นี้ถ้า Word เป็น 2 จะขึ้นบรรทัดใหม่ สำหรับค่าเริ่มต้นกำหนดให้มีค่าเท่ากับ 0
- (2) เป็นคำสั่งให้ทำงานวนรอบเพื่ออ่านอักขระทีละตัวอักษร ในกรณีที่ยังไม่พบ Ctrl-z จะกระทำงานวนรอบไปเรื่อยๆ
- (3) อ่านอักขระหนึ่งตัวอักษรจากข้อความที่ผู้ใช้ป้อน
- (4) ตรวจสอบค่าของข้อมูลที่อ่านเข้ามากกรณีที่เป็นช่องว่าง แสดงว่าในที่นี้หมดคำหนึ่งคำ เพราะช่องว่างจะอยู่ระหว่างคำเสมอ จะกำหนดให้ค่า N มีค่าเป็น 0 เพื่อนับจำนวนตัวอักษรของคำต่อไป ต่อจากนั้นตรวจสอบว่าการพิมพ์ในบรรทัดครบ 2 คำหรือยังในกรณีที่ครบแล้วคือ Word มีค่าเท่ากับ 2 จะขึ้นบรรทัดใหม่ และกำหนดค่าเริ่มต้นของจำนวนคำในบรรทัดใหม่ ให้มีค่าเท่ากับ 0
- (5) กรณีที่เป็นตัวอักษรที่ไม่ใช่ช่องว่าง จะเพิ่มตัวนับของคำขึ้น 1 ต่อจากนั้นตรวจสอบว่าเป็นคำใหม่หรือไม่ การตรวจสอบโดยการตรวจสอบว่าถ้า N มีค่าเท่ากับ 1 แสดงว่าเป็นอักษรตัวแรกของคำ ให้เพิ่มค่า Word ขึ้น 1
- (6) เป็นคำสั่งในการพิมพ์ตัวอักษรทางจอภาพ ซึ่งในที่นี้จะพิมพ์ทีละตัวอักษรตามคำที่อ่านมาการพิมพ์จะถูกควบคุมโดยคำสั่งในตอนต้น

แบบฝึกหัดท้ายบทที่ 10 ข้อ 4

จงเขียนโปรแกรมรับเลขฐาน 10 ทางแป้นพิมพ์ จงแปลงให้เป็น ฐาน 2, ฐาน 8, ฐาน 16

Program Change_Number (Input,Output);

Type Structure = Array[1..8] of Char;

Var A : Structure ;

Base , X , I ,J , Menu : Integer;

Procedure Convert(Base : integer; B:integer; VAR A : Structure) ;(1)

Var Temp ,I : Integer;

Begin

I := 0;(2)

While B <> 0 do(3)

Begin

I := I + 1;(4)

Temp := B mod Base ;(5)

Case Temp of(6)

0 : A[I] := '0';

1 : A[I] := '1';

2 : A[I] := '2';

3 : A[I] := '3';

4 : A[I] := '4';

5 : A[I] := '5';

6 : A[I] := '6';

7 : A[I] := '7';

8 : A[I] := '8';

9 : A[I] := '9';

```

10 : A[I] := 'A';
11 : A[I] := 'B';
12 : A[I] := 'C';
13 : A[I] := 'D';
14 : A[I] := 'E';
15 : A[I] := 'F';

End;
B := B div Base; .....(7)
End;
End;
Begin { Main}
Repeat .....(8)
  WriteLn('โปรดเลือกการทำงานต่อไปนี้'); .....(9)
  WriteLn('1. แปลงเลขฐาน10 เป็นเลขฐาน 2');
  WriteLn('2. แปลงเลขฐาน10 เป็นเลขฐาน 8');
  WriteLn('3. แปลงเลขฐาน10 เป็นเลขฐาน 16');
  WriteLn('4 จบการทำงาน');
  Write('เลือก 1 หรือ 2 หรือ 3 หรือ 4 ?');readLn(Menu); .....(10)
  Case Menu of .....(11)
    1 : Base := 2;
    2 : Base := 8;
    3 : Base := 16;
  End;
  Write('กรุณาป้อนเลขฐาน 10 ที่ท่านต้องการแปลง') ;readLn(X);
  Convert(Base , X , I , A); .....(12)
  Write('ผลลัพธ์ของการแปลงเป็นฐาน',Base,'มีค่าดังนี้');
  For J := I downto 1 do .....(13)

```

Write(A[J]);

ReadIn;

Until Menu=4 ;(14)

End.

คำอธิบายโปรแกรม

- (1) เป็นโพรซีเจอร์ในการเปลี่ยนเลขฐาน 10 เป็นเลขฐาน 2 หรือ 8 หรือ 16 โดยการเรียกใช้ ต้องส่ง เลขฐานที่ต้องการเปลี่ยน(Base) ค่าของข้อมูลฐาน 10 (B) และตัวแปรที่อ้างอิง ในการเก็บผลลัพธ์เป็นอาเรย์ขนาดความยาว 8 ตัวอักษร(A)
- (2) ในการแปลงใช้วิธีนำค่าฐานหารเลขฐานสิบ เศษที่ได้จะเก็บในอาเรย์ A ช่องที่ i ก่อน การปฏิบัติการกำหนดค่าเริ่มต้นให้ i มีค่าเท่ากับ 0
- (3) การหารนั้นจะตั้งหารและนำเศษไปเก็บในตัวแปรผลลัพธ์ไปเรื่อยๆ จนกระทั่งเลขที่เป็นตัว ตั้งมีค่าเป็น 0 ดังนั้นการทำงานจะกระทำวนรอบจนกระทั่ง B เป็น 0
- (4) เพิ่มค่า i ขึ้นหนึ่งเพื่อเป็นช่องของข้อมูลที่จะนำเศษที่ได้มาเก็บ
- (5) นำค่าเลขฐาน 10 ที่ส่งมาหารด้วยค่าของฐานที่ต้องการแปลง ผลที่ได้เป็นเศษเก็บในตัวแปร Temp
- (6) แปลงเศษที่ได้เป็นอักขระ ในกรณีที่เป็นฐาน 16 ต้องแปลงให้เป็นค่า A หรือ B หรือตัว อื่นๆ นำอักขระที่แปลงได้เก็บในตัวแปรผลลัพธ์ช่องที่ i
- (7) เมื่อได้เศษที่ได้ตัวแรกแล้ว นำเลขจำนวนเต็มที่หารได้ที่เหลือ เก็บในตัวแปร B ซึ่งจะ ทำงานในรอบต่อไป (3) การทำงานในรอบต่อไปจะเพิ่มค่า i และนำเศษที่ได้แปลงเป็น อักขระเก็บในตัวแปรผลลัพธ์และลดค่า B ไปเรื่อยๆจนกระทั่ง B มีค่าเท่ากับ 0 จะหยุด การทำงานวนรอบผลที่ได้ จะได้เศษที่ได้จากการหารเก็บในตัวแปร A จำนวนเท่ากับ i ค่า
- (8) การทำงานของ Main Program จะทำงานวนรอบให้ผู้เลือกใช้การทำงาน
- (9) เป็นเมนูการทำงานให้ผู้เลือกใช้
- (10) รับค่าที่ผู้ใช้เลือกตามเมนู เก็บในตัวแปร Menu
- (11) นำค่าที่ผู้ใช้เลือกมากำหนดค่าให้กับ Base ซึ่งเก็บค่าฐานที่ผู้ใช้ต้องการแปลง

- (12) ต่อจากนั้นให้ผู้ใช้ป้อนเลขฐาน 10 ที่ต้องการแปลงทางแป้นพิมพ์ เรียกใช้โปรแกรมในการแปลงค่าทำงานโดยส่ง ฐานที่ต้องการแปลง ค่าของเลขฐาน 10 ไปยังโปรแกรม และกำหนดตัวแปรอ้างอิงในการเก็บผลลัพธ์จากการแปลงเลขฐานด้วยในที่นี้คือตัวแปร A ซึ่งก่อนเรียกใช้ตัวแปร A ยังไม่มีค่า แต่เมื่อการทำงานของโปรแกรมสิ้นสุดลง ตัวแปร A จะเก็บเศษที่ได้จากการหารในช่องที่ 1, 2, ... โดยเก็บได้สูงสุด 8 ช่อง ในที่นี้ข้อมูลผลลัพธ์จะเก็บได้ 1 ช่องเพราะเราใช้ตัวแปร 1 ในการนับช่องที่เก็บเศษที่ได้จากการหาร
- (13) เป็นการพิมพ์ผลลัพธ์จากการแปลงค่าเลขฐาน 10 เป็นเลขฐานที่ต้องการ โดยการพิมพ์จะพิมพ์จากตำแหน่งที่ 1 ย้อนไปถึงตำแหน่งที่ 1
- (14) เป็นเงื่อนไขในการทำงานวนรอบ ผู้ใช้ต้องกด 4 เท่านั้นการทำงานจึงสิ้นสุด

แบบฝึกหัดท้ายบทที่ 10 ข้อ 5

จงเขียนโปรแกรมเรียงลำดับข้อมูลที่ต้องการโดยเขียนเป็น โพรซีเจอร์

```
Program Data_sort (Input , Output);  
Type  
    Structure = array[1..100] of integer;  
Var  
    I ,N : Integer;  
    Data : Structure;  
Procedure  Sorting( N : Integer ;Var Data : Structure );      .....(1)  
Var    I , J , T : Integer ;  
Begin  
    For I := 1 to N-1 Do      .....(2)  
        For J := I+1 to N do      .....(3)  
            If Data[I]>Data[J] then      .....(4)  
                Begin  
                    T      := Data[I] ;  
                    Data[I] := Data[J];  
                    Data[J] := T ;  
                End;  
    End;  
End;  
Begin {Main}  
    Write ( ' ข้อมูลมีทั้งหมด = ' );    Readln (N);  
    For I := 1 to N Do      .....(5)  
        Read (Data[ I ] );  
    Sorting (N , Data);      .....(6)
```

For I := 1 to N do

.....(7)

Write (Data[I] :5);

End.

คำอธิบายการทำงานของโปรแกรม

- (1) เป็นโพรซีเจอร์ในการเรียงลำดับโดยส่งข้อมูลมายังโพรซีเจอร์นี้ 2 ค่าคือจำนวนข้อมูลทั้งหมดและกลุ่มของข้อมูลที่ต้องการเรียงลำดับ
- (2) การทำงานนี้จะเรียงลำดับข้อมูลจากน้อยไปมากโดยทำงานวนรอบทั้งหมด $n-1$ รอบ
- (3) เป็นการทำงานวนรอบซึ่งเท่ากับ $n-1+1$ รอบ
- (4) เป็นเงื่อนไขในการตรวจสอบค่า การทำงานนี้จะทำการเปรียบเทียบค่าของข้อมูลตำแหน่งที่ i กับข้อมูลตำแหน่งที่ถัดไป จนถึงตัวสุดท้ายเพื่อหาค่าที่น้อยที่สุด นำไปเก็บไว้ในตำแหน่งที่ i การทำงานจะเปรียบเทียบไปเรื่อยๆจนกระทั่ง i เป็นตัวที่ $N-1$ จะเปรียบเทียบกับข้อมูลตัวสุดท้าย จะได้ข้อมูลเรียงลำดับจากน้อยไปมากดังที่ต้องการ
- (5) อ่านข้อมูลจำนวน N ตัวเก็บในตัวแปรอาเรย์ Data
- (6) เรียกใช้โพรซีเจอร์ Sorting เพื่อทำการเรียงลำดับข้อมูลจากน้อยไปมาก
- (7) พิมพ์ข้อมูลที่เรียงลำดับในตำแหน่งที่ 1 ถึงที่ N ออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 11

แบบฝึกหัดท้ายบทที่ 11 ข้อ 1

เขียนโปรแกรมเก็บเบอร์โทรศัพท์ เป็นอาเรย์ของเรคอร์ดประกอบด้วยฟิลด์ชื่อ ฟิลด์รหัสทางไกล ฟิลด์หมายเลขโทรศัพท์ ให้โปรแกรมสามารถค้นหารหัสทางไกล และหมายเลขโทรศัพท์ โดยป้อนชื่อที่ต้องการทางแป้นพิมพ์

```
Program Find_Telephones;
```

```
Type
```

```
String10 = String[10] ;
```

```
Telephone = record
```

```
    Name : String10;
```

```
    Code : string[3];
```

```
    Number: string[7];
```

```
End;
```

```
    ARRAYTelephone = ARRAY[1..100] of Telephone;
```

```
Var
```

```
Phone : ARRAYTelephone; .....(1)
```

```
A : Telephone .....(2)
```

```
I : integer;
```

```
N : String10;
```

```
Procedure Get_Data(I : integer ; VAR Phone : ARRAYTelephone); .....(3)
```

```
Var K : integer;
```

```
Begin
```

```
    For K := 1 to I do
```

```
        With Phone[K] do
```

```
            Begin
```

```
                Write('ชื่อ'); readln(Name);
```

```
                Write('รหัสทางไกล'); readln(Code);
```

```
                Write('หมายเลขโทรศัพท์');readln(Number);
```

```
            End;
```

```
End;
```

```

Procedure Find(N:String10 ; I : Integer ; Phone :ARRAYTelephone; .....(4)
                                VAR A : Telephone);

```

```

VAR   K : Integer;

```

```

Begin

```

```

    A.Name := '';

```

```

    For K := 1 to I do

```

```

        If N= Phone.Name then

```

```

            A :=Phone[K]

```

```

End;

```

```

Procedure Write_data(A:Telephone); .....(5)

```

```

Begin

```

```

    WITH A do

```

```

        WriteLn('ชื่อ ',Name,' รหัสทางไกล ',Code ,' หมายเลขโทรศัพท์', Number)

```

```

End;

```

```

Begin {Main}

```

```

    Write('จำนวนของข้อมูล');readLn(I);

```

```

    Get_Data(I , Phone); .....(6)

```

```

    Write('ชื่อที่ต้องการค้นหา');readLn(N);

```

```

    Find(N, I ,Phone, A); .....(7)

```

```

    If A.Name=N then Write_Data(A) .....(8)

```

```

    Else WriteLn('ไม่มีข้อมูล');

```

```

End.

```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร Phone เป็นอาเรย์ของเรคอร์ดเพื่อเก็บรายละเอียดของข้อมูลได้สูงสุด 100 คน
- (2) ประกาศตัวแปร A เป็นโครงสร้างข้อมูล Telephone

- (3) เป็นโพรซีเจอร์ที่ทำหน้าที่รับข้อมูลชื่อ รหัสทางไกล และหมายเลขโทรศัพท์ของบุคคลใดๆ จำนวน 1 คนทางแป้นพิมพ์เก็บในตัวแปร Phone
- (4) เป็นโพรซีเจอร์ในการค้นหาข้อมูลโดยเปรียบเทียบชื่อของบุคคลที่ต้องการค้นหากับข้อมูลที่เก็บอยู่ในตัวแปร Phone โดยเริ่มค้นหาตั้งแต่ตัวที่ 1 ถึงตัวที่ 1 กรณีที่พบจะคัดลอกข้อมูลตำแหน่งที่พบให้แก่ตัวแปร A ในกรณีที่ไม่พบตัวแปร A ในฟิลด์ Name มีค่าเป็นช่องว่าง
- (5) เป็นโพรซีเจอร์ในการพิมพ์ค่าที่เก็บในตัวแปร A ออกทางจอภาพโดยใช้คำสั่ง With do
- (6) เรียกใช้โพรซีเจอร์ Get_Data
- (7) เรียกใช้โพรซีเจอร์ Find
- (8) เป็นคำสั่งเงื่อนไขในการตรวจสอบว่าข้อมูลที่ต้องการค้นหา มีหรือไม่โดยตรวจสอบจากข้อมูลที่เก็บในตัวแปร A ฟิลด์ชื่อ ในกรณีที่เป็นช่องว่างแสดงว่าค้นหาไม่พบ แต่ถ้าไม่ใช่เรียกใช้โพรซีเจอร์ Write_data

แบบฝึกหัดท้ายบทที่ 11 ข้อ 2

ห้องสมุดแห่งหนึ่งต้องการเก็บข้อมูลต่อไปนี้เพื่อประโยชน์ในการเก็บหนังสือและให้ยืม ประกอบด้วย

- reference number
- title
- number of author/composer
- name of publisher
- class of item-book or recording
- number of edition (book only)
- year of publication (books only)
- name of performer (recordings only)

Program libraries;

Type

```
Notebook    =    record
                Ref_no    : String[4];
                Title     : String[10];
                No_au     : Integer;
                Publisher  : String[30];
                Class     : String[3];
                No_edit   : Integer;
                Year      : 1980..2020;
                Name      : String[20];
```

End;

Var

```
Books : Array[1..1000] of Notebook;
I      : 1..1000;
```

Begin

```
For I:=1 to 1000 do
```

```
    With Books[I] do
```

```
        Begin
```

```
            Write('reference number=');readln(ref_no);
```

```
            Write('title=');readln(title);
```

```
            Write('number of author/composer=');readln(no_au);
```

```
            Write('name of publisher=');readln(publisher);
```

```
            Write('class of item-book or recording=');readln(class);
```

```
            Write('number of edition=');readln(no_edit);
```

```
            Write('year of publication=');readln(year);
```

```
            Write('name of performer=');readln(name);
```

End;

End.

คำอธิบายโปรแกรม

โปรแกรมนี้เป็นการรับข้อมูลซึ่งเป็นข้อมูลของหนังสือในการเก็บและการยืม โดยป้อนข้อมูลทั้งหมด 1000 เรคอร์ดเก็บในตัวแปร Books

แบบฝึกหัดท้ายบทที่ 11 ข้อ 3

เขียนโปรแกรมค้นหาห้องว่างในคอนโดมิเนียมแห่งหนึ่ง โดยเก็บรายละเอียดของห้องพักให้เหมาะสม

Program Rent;

Type

Apartment = record

Room : String[3]; {หมายเลขห้อง}

Rent : integer; {ค่าเช่า}

Flag : 0..1; {0 = ว่าง 1 = มีคนเช่า }

Name : String[20]; {ชื่อผู้เช่า}

Address : String[50]; {ที่อยู่ของผู้เช่า}

End;

Apartments = Array[1..100] of Apartment;

Var

A : Apartments;

I : 1..100;

Begin

For I:=1 to 100 do(1)

With Apartments[I] do

Begin

```

Write('room=');readln(Room);
Write('Rent=');readln(Rent);
Write('flag=');readln(Flag);
If flag = 1 then .....(2)
    Begin
        Write('Name=');readln(Name);
        Write('Address=')readln(Address);
    End;
End;
Write('ห้องที่ว่างได้แก่');
For I := 1 to 100 do .....(3)
    With Apartments[I] do
        If Flag=0 then Write(Room:5);

```

End.

คำอธิบายโปรแกรม

- (1) เป็นคำสั่งการทำงานวนรอบทั้งหมด 100 รอบเพื่อรับรายละเอียดของห้องพัก ประกอบด้วยหมายเลขห้องพักกำหนดไว้สูงสุด 1000 ห้อง ค่าเช่า สถานะของห้อง ชื่อผู้เช่า และที่อยู่
- (2) การป้อนข้อมูลไม่จำเป็นต้องป้อนทุกฟิลด์ขึ้นอยู่กับสถานะของห้องพัก กรณีที่ห้องพักว่าง ก็ไม่จำเป็นต้องป้อนรายละเอียดผู้เช่า ในกรณีที่ป้อนทุกฟิลด์แสดงว่าห้องพักห้องนั้นมีคนอยู่
- (3) การค้นหาว่าห้องพักใดว่าง เพียงแต่ตรวจสอบในฟิลด์ Flag ว่ามีค่าเป็น 0 หรือไม่เท่านั้น สำหรับคำสั่งนี้จะตรวจสอบในฟิลด์นี้ทุกเรคอร์ดที่ป้อน เพื่อพิมพ์เฉพาะหมายเลขห้องที่ว่างเท่านั้นออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 11 ข้อ 4

เขียนโปรแกรมเก็บคะแนนสอบวิชาคณิตศาสตร์ สถิติ วิทยาศาสตร์ คอมพิวเตอร์ ของนักศึกษา N คน จงนำคะแนนแต่ละคนมาหาคะแนนเฉลี่ยของแต่ละวิชา และจำนวนของนักเรียนที่สอบผ่านกรณีคะแนนมากกว่าหรือเท่ากับ 50 คะแนน โดยเก็บเป็นอาเรย์ของเรคอร์ด

Program grade;

Type

Total = 0..100

Exam = record

Math : Total;

Stat : Total;

Science : Total;

Computer: Total;

End;

Var

Ex : array[1..200] of Exam;

I, N : integer;

Aver_m ,Aver_st , Aver_sc, Aver_com : real;

Count1 ,Count2, Count3, Count4 : integer;

Sum1, Sum2, Sum3, Sum4 : integer;

Begin

Sum1:=0 ;Sum2:=0; Sum3:=0; Sum4:=0;(1)

Count1:=0; Count2:=0; Count3:=0; Count4:=0;

Write('จำนวนของนักศึกษาทั้งหมดเท่ากับ');readln(N);

For I:=1 to N do(2)

Begin

With Ex[I] do(3)

```

Begin
    Write('Math=');readln(Math);
    Sum1:=Sum1+Math; .....(4)
    If Math>=50 then Count1:=Count1+1;
    Write('stat=');readln(Stat);
    Sum2:=Sum2+Stat; .....(5)
    If Stat>=50 then Count2:=Count2+1;
    Write('Science=');readln(Science);
    Sum3:=Sum3+Science;
    If Science>=50 then Count3:=Count3+1;
    Write('computer=');readln(Computer);
    Sum4:=Sum4+Computer;
    If Computer>=50 then Count4:=Count4+1;
End;
End;
Aver_m:=Sum1/n; .....(6)
Aver_st:=Sum2/n;
Aver_sc:=Sum3/n;
Aver_com:=Sum4/n;
Writeln('Average Math=',Aver_m:5:2,'Pass=',Count1); .....(7)
Writeln('Average Stat=',Aver_st:5:2,'Pass=',Count2);
Writeln('Average Science=',Aver_sc:5:2,'Pass=',Count3);
Writeln('Average Computer=',Aver_com:5:2,'Pass=',Count4);
End.

```

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปรที่ใช้ในการนับ และเก็บผลรวมทั้งหมดให้มีค่าเท่ากับ 0 โดยในโปรแกรมนี้อัปเดตคะแนนสอบ 4 วิชาด้วยกัน ดังนั้นจึงมีตัวแปรสำหรับนับคนที่สอบผ่าน ผลรวมของคะแนนแต่ละวิชาเพื่อนำไปหาค่าเฉลี่ย อย่างละ 4 ตัวแปรด้วยกัน
- (2) รับจำนวนนักศึกษาทั้งหมดทางแป้นพิมพ์ในขั้นสูงสุดได้ 200 คน ดังนั้นสามารถใช้คำสั่ง for..do สำหรับให้ทำงานวนรอบได้ทั้งหมด N รอบ
- (3) เป็นการใช้คำสั่ง With..do เพื่ออ้างถึงฟิลด์สำหรับตัวแปรชนิดอาเรย์ Ex ได้สะดวก
- (4) การรับข้อมูลเริ่มจากการป้อนคะแนนของวิชาคณิตศาสตร์ของนักศึกษาคนที่ 1 ซึ่งในที่นี้เรานำคะแนนที่รับมาเก็บในตัวแปร Sum1 และตรวจสอบค่าที่รับเลยว่าสอบผ่านหรือไม่ ถ้าสอบผ่านจะนับเพิ่ม 1 ในตัวแปร Count1
- (5) ต่อจากนั้นทำงานในลักษณะเดียวกันกับข้อ (4) แต่เปลี่ยนเป็นวิชาสถิติ วิชาวิทยาศาสตร์ และคอมพิวเตอร์ และเก็บในตัวแปรที่มีชื่อต่าง ๆ กัน การทำงานจะทำงานวนรอบทั้งหมดจำนวน N รอบ โดยกลับไปทำที่(2)
- (6) เมื่อรับข้อมูลครบ N คนแล้ว ผลของการทำงานในขณะนี้คือจะได้ค่าของข้อมูล Sum1, Sum2, Sum3, Sum4 และ Count1, Count2, Count3, Count4 เรียบร้อยแล้วซึ่งตัวแปร Sum จะเก็บค่าผลบวกทั้งหมดของวิชาทั้งสี่ และตัวแปร Count จะเก็บจำนวนนักศึกษาที่สอบได้มากกว่าหรือเท่ากับ 50 คะแนนของแต่ละวิชาตามลำดับ เราสามารถหาค่าเฉลี่ยของคะแนนแต่ละวิชาโดยนำค่า N มาหาร เก็บในตัวแปร Average ของแต่ละวิชา
- (7) พิมพ์ผลลัพธ์จากการทำงานออกทางจอภาพ

แบบฝึกหัดท้ายบทที่ 12

แบบฝึกหัดท้ายบทที่ 12 ข้อ 1

จากตัวอย่างที่ 12.3 เป็นการสร้างไฟล์เพื่อเก็บ ชื่อ อายุ และเงินเดือน ให้ทำการค้นหาข้อมูลของบุคคลใดๆ เพื่อแก้ไขข้อมูลในเรคคอร์ดนั้น

Program ex1;

```
Type String10 = String[10];
```

```
Person = record
```

```
    Name : String10 ;
```

```
    Age   : Integer;
```

```
    Salary : real;
```

```
End;
```

```
Var fp : file of Person;
```

```
    Data : Person;
```

```
    N : String10;
```

```
    Found : Boolean;
```

```
Begin
```

```
    Found := false; .....(1)
```

```
    Write('ชื่อของบุคคลที่ต้องการแก้ไขข้อมูลคือ'); readln(N); .....(2)
```

```
    Assign(fp, 'person.dat'); .....(3)
```

```
    Reset(fp); .....(4)
```

```
    While (not EOF(fp)) and (Not Found) do .....(5)
```

```
        Begin
```

```
            Read(fp,Data); .....(6)
```

```
            If N=Data.Name then .....(7)
```



```

Begin
    Found := true;          .....(8)
    Seek(fp,FILEPOS(fp)-1); .....(9)
    Data.Name := N ;
    Writeln('ชื่อ',N);      .....(10)
    Write('อายุใหม่'); readln(Data.Age);
    Write('เงินเดือนใหม่'); readln(Data.Salary);
    Write(fp,Data);        .....(11)
End;
End;
Close(fp);                .....(12)
End.

```

คำอธิบายโปรแกรม

- (1) กำหนดให้ตัวแปร Found มีค่าเท่ากับ False โดยตัวแปรนี้ทำหน้าที่ตรวจสอบการค้นหาข้อมูลที่ต้องการแก้ไขกล่าวคือถ้าไม่พบจะมีค่าเป็น False แต่ถ้าพบข้อมูลจะถูกเปลี่ยนเป็น True
- (2) รับชื่อของบุคคลที่ต้องการแก้ไขข้อมูลในตัวแปร N
- (3) กำหนดตัวแปรไฟล์ชี้ไปที่แฟ้ม person.dat
- (4) เปิดแฟ้มข้อมูลในตัวแปรไฟล์ fp ชี้อยู่โดยเลื่อนไปที่เรคอร์ดข้อมูลตัวแรก เพื่ออ่านข้อมูลทำงาน
- (5) ทำงานวนรอบในขณะที่ยังไม่หมดข้อมูล และยังไม่พบข้อมูลที่ต้องการแก้ไข (6)
- (6) อ่านข้อมูลจากแฟ้ม 1 เรคอร์ดเก็บในตัวแปร Data
- (7) ตรวจสอบว่าใช่เรคอร์ดที่ต้องการหรือไม่ ถ้าใช่ทำ (8) ถ้าไม่ใช่ (5)
- (8) ถ้าใช่ข้อมูลที่ต้องการพบ เปลี่ยนค่าตัวแปร Found เป็น True
- (9) เลื่อนให้ตัวชี้ข้อมูลในไฟล์เลื่อนขึ้นไป 1 เรคอร์ด
- (10) ทำการอ่านข้อมูลใหม่ที่ต้องการแก้ไข ซึ่งเป็นอายุใหม่ หรือ เงินเดือนใหม่เก็บในตัวแปร Data ในฟิลด์อายุ และเงินเดือนตามลำดับ

- (11) บันทึกข้อมูลใหม่ลงในตำแหน่งของเรคอร์ดที่ตัวชี้ชี้อยู่ มีผลให้ข้อมูลใหม่แทนที่ข้อมูลเก่า
- (12) เมื่อเงื่อนไขการทำงานวนรอบเป็นเท็จ ซึ่งอาจแก้ไขข้อมูลที่ต้องการแล้ว หรือในกรณีที่ไม่พบข้อมูลที่ต้องการแก้ไขก็ตาม ปิดเพิ่ม เป็นอันจบโปรแกรม

แบบฝึกหัดท้ายบทที่ 12 ข้อ 2

เขียนโปรแกรมสร้าง Text file และจงเพิ่มข้อมูลในไฟล์นี้ได้ด้วย

Program Append_Text ;

Var fp : Text;(1)

Ch : Char;

Begin

Assign(fp, 'text.dat');

Append(fp);(2)

While not eof do(3)

Begin

Read(ch);(4)

Write(fp,ch);(5)

End;

Close(fp);

End.

คำอธิบายโปรแกรม

- (1) ประกาศตัวแปร fp เป็นชนิดไฟล์ตัวอักษร
- (2) เป็นการเปิดไฟล์ชนิดตัวอักษรเพื่อเพิ่มเติมข้อมูลในแฟ้มนั้น ตัวชี้ไฟล์จะเลื่อนไปยังท้ายไฟล์
- (3) ผู้ใช้จะป้อนข้อมูลทางแป้นพิมพ์ไปเรื่อยๆ โดยสิ้นสุดด้วย Ctrl-z การทำงานจะทำงานวนรอบ จนกระทั่งพบ Ctrl-z
- (4) การทำงานใน loop จะทำการอ่านข้อมูลที่ผู้ใช้ป้อนทีละ 1 ตัวอักษรเก็บในตัวแปร Ch

- (5) บันทึกข้อมูลที่เก็บในตัวแปร Ch ลงแฟ้ม ที่ละตัวจนกระทั่งหมดข้อมูล ปิดแฟ้มเป็นอันจบโปรแกรม

แบบฝึกหัดท้ายบทที่ 12 ข้อ 3

เขียนโปรแกรมเก็บผลการลงทะเบียนของนักศึกษา N คน เช่นเก็บชื่อ นามสกุล รหัสประจำตัว รหัสวิชา เกรดของวิชานั้นๆ จำนวนเครดิต โดยนักศึกษา 1 คนลงได้ไม่เกิน 24 หน่วยกิต ให้นำเกรดเฉลี่ยของนักศึกษาแต่ละคน

Program Coumpute_Grade;

Type

Description = record(1)

Subject : String[5];

Credit : Integer;

Grade : Char;

End;

Person = record(2)

Name : String[20];

Code : String[10];

N_Subject : Integer;

Subjects : Array[1..8] of Description ;

Average : real;

End;

VAR

fp : file of Person

Student : Person;

N , I , J , T : integer;

Sum : Real;

Begin

Assign (fp,'Data.dat');(3)

Rewrite(fp);

Write('นักศึกษาทั้งหมดจำนวนทั้งหมด');readln(N);

For I := 1 to N do(4)

With Student do(5)

Begin

Write('ชื่อ-นามสกุล');readln(Name);

Write('รหัสประจำตัว');readln(Code);

Write('จำนวนวิชาที่ลงทะเบียน');readln(N_Subject);(6)

T := 0 ; Sum := 0.0;(7)

For J := 1 to N_Subject do(8)

With Subjects[J] do(9)

Begin

Write('รหัสวิชาที่ลงทะเบียน');Readln(Subject);

Write('ก่ีเครดิต');readln(Credit);

Write('เกรดที่ได้');readln(Grade);

Case Grade of(10)

'p' , 'P' : begin

T := T + Credit;

Sum := Sum+2.25*Credit;

end;

'g' , 'G' : begin

T := T + Credit;

Sum := Sum+4*Credit;

end;

End; {CASE}

End; {FOR J}

Average := Sum / T ;(11)

Write(fp,Student);(12)

End; {With..do}

Close(fp);(13)

End.

คำอธิบายโปรแกรม

การทำงานของโปรแกรมนี้อาศัยการคิดเกรดของมหาวิทยาลัยรามคำแหง กล่าวคือนักศึกษาที่ได้เกรด P ได้คะแนนเท่ากับ 2.25 สำหรับเกรด G ได้คะแนนเท่ากับ 4.0 สำหรับ F ทางมหาวิทยาลัยไม่นำมาคิด การทำงานของโปรแกรมนี้อาศัยข้อจำกัดที่ว่านักศึกษา 1 คนลงได้ไม่เกิน 24 หน่วยกิต ดังนั้นจึงกำหนดอาเรย์ในการเก็บรายละเอียดของวิชาที่นักศึกษาลงทะเบียน 8 วิชาด้วยกัน ข้อมูลของนักศึกษาแต่ละคนจะเก็บบันทึกลงเพิ่มข้อมูลชื่อ data.dat

- (1) เป็นส่วนของการกำหนดรายละเอียดของวิชาที่นักศึกษาลงทะเบียนประกอบด้วยรหัสวิชา จำนวนเครดิต และเกรดที่นักศึกษาได้
- (2) เป็นการกำหนดระเบียบที่เก็บข้อมูลของนักศึกษาประกอบด้วย 5 ฟิลด์ด้วยกันคือ ชื่อ รหัสประจำตัว จำนวนวิชาที่ลงทะเบียน รายละเอียดของวิชาที่ลงทะเบียนทั้งหมดเป็นอาเรย์สูงสุดได้ 8 วิชา และเกรดเฉลี่ยที่ได้
- (3) เป็นการกำหนดตัวแปรไฟล์ให้ชี้ไปที่เพิ่ม data.dat เพื่อบันทึกข้อมูลใหม่ลงในแฟ้มนี้
- (4) รับจำนวนนักศึกษาที่ต้องการเก็บข้อมูลทั้งหมดทางแป้นพิมพ์เก็บในตัวแปร N ใช้คำสั่งทำงานวนรอบเพื่อรับข้อมูล N รอบ
- (5) อ่างตัวแปรชนิดเรคอร์ดโดยใช้คำสั่ง With..do
- (6) รับชื่อ รหัสประจำตัว และจำนวนวิชาที่ลงทะเบียนทางแป้นพิมพ์
- (7) กำหนดค่าเริ่มต้นให้แก่ตัวแปรที่นับจำนวนหน่วยกิตทั้งหมดที่นักศึกษาสอบผ่าน และผลรวมของเกรดที่ได้ให้มีค่าเริ่มต้นเป็น 0 และ 0.0 ตามลำดับ
- (8) ใช้คำสั่งทำงานวนรอบเพื่อรับรายละเอียดของวิชาต่างๆที่ได้ลงทะเบียนรวมทั้งเกรดที่ได้เท่ากับจำนวนวิชาที่ลงทะเบียน

- (9) ใช้คำสั่ง with..do เพื่ออ้างถึงตัวแปรชนิดเรคคอร์ด รับรหัสวิชาที่ลงทะเบียนเรียน จำนวนเครดิต และเกรดที่ได้ ทางแป้นพิมพ์
- (10) นำเกรดที่ได้มาทำงานตามเงื่อนไข กรณีที่ได้ 'F' ไม่นำมาคิด ซึ่งการทำงานจะวนรอบเพื่อรับข้อมูลวิชาที่นักศึกษาลง นำเกรดมาคำนวณตามเงื่อนไขจนครบทุกกระบวนวิชา จะหลุดจากการทำงานไปทำ (11)
- (11) คำนวณหาเกรดเฉลี่ยของนักศึกษาเก็บใน Student.Average
- (12) บันทึกข้อมูล Student ในแฟ้ม data.dat การทำงานจะวนรอบเพื่อทำงานซ้ำ (4)-(12) จนกระทั่งหมดครบทุกคน
- (13) ปิดแฟ้ม

แบบฝึกหัดท้ายบทที่ 13

แบบฝึกหัดท้ายบทที่ 13 ข้อ 1

จงเขียนโปรแกรมเพื่อตรวจสอบหมายเลขของการเล่นบิงโก โดยการเล่นนี้ผู้เข้าแข่งขันต้องทายหมายเลข 4 ตัว โดยไม่ซ้ำกันให้ถูกต้องโดยให้ผู้เข้าแข่งขันทายได้เพียง 3 ครั้งเท่านั้น กรณีที่หมายเลขตรงกับที่กรรมการตั้งไว้โดยไม่คำนึงถึงลำดับแสดงว่าได้รางวัล

Program Ex13_1;

VAR Bingo : Set of Integer;(1)

I , N, J : Integer;

Correct , Continue : Boolean;

Begin

Bingo := [2 , 4 , 9 , 7]; {เป็นหมายเลขที่กรรมการตั้งไว้}

I := 1 ; Correct := False;(2)

While (I<=3) and (not Correct) do(3)

Begin

```

Continue := True; J := 1; .....(4)
While (J<=4) and Continue do .....(5)
    Begin
        Write('กรุณาทายเลขตัวที่',J);readln(N); .....(6)
        If N IN Bingo then J := J+1 .....(7)
        Else
            Begin .....(8)
                Writeln('คุณทายเลขผิดแล้วค่ะ');
                Continue := False;
            End;
        End; {While J}
        If (J>4) and Continue then .....(9)
            Correct := True;
        Else
            Begin
                Write('คุณทายตัวเลขในรอบที่',I,'ผิด ทายใหม่นะคะ');
                I := I+1; .....(10)
            End;
        End; {While I}
        If Correct then Writeln('คุณทายตัวเลขถูกต้อง') .....(11)
        Else Writeln('เสียใจด้วยนะคะที่ทายไม่ถูกเลย');
    End.

```

คำอธิบายโปรแกรม

- (1) เป็นการประกาศตัวแปร Bingo เป็นข้อมูลชนิดเซตของเลขจำนวนเต็ม
- (2) นำค่าเลขจำนวนเต็มใดๆ 4 ตัวที่แตกต่างกันเก็บในตัวแปร Bingo การทำงานจะให้ผู้ใช้โปรแกรมทายตัวเลขทั้งสี่ตัว เป็นจำนวน 3 ครั้ง กำหนดตัวแปร I เพื่อนับจำนวนครั้งที่ผู้

- ใช้ทายกำหนดให้ค่าเริ่มต้นมีค่าเท่ากับ 1 สำหรับตัวแปร Correct เป็นตัวแปรที่เก็บค่าทางตรรก ในกรณีที่ทายเลขทั้งสี่ตัวไม่ถูกต้องค่าของตัวแปรนี้มีค่าเท่ากับ False
- (3) การทำงานในที่นี้จะทำงานวนรอบขึ้นอยู่กับค่าของ I และ Correct กล่าวคือค่า I ต้องมีค่าน้อยกว่าและเท่ากับสาม และ ค่าของ Correct ต้องเป็น False เท่านั้น คือยังไม่ครบการทาย 3 ครั้งและยังทายไม่ถูก ในกรณีที่ครบสามครั้งยังทายไม่ถูก หรือ ยังไม่ครบสามครั้งแต่ทายถูกก่อน ก็มีผลให้การทำงานของโปรแกรมหยุดการทำงานวนรอบทันที
 - (4) การทายแต่ละครั้งผู้ใช้โปรแกรมต้องทายตัวเลข 4 ตัวด้วยกัน ดังนั้นจึงมีการกำหนดตัวแปร J ให้นับจำนวนตัวเลขที่ผู้ใช้โปรแกรมทาย และตัวแปร Continue ใช้สำหรับตรวจสอบว่าผู้ใช้ทายตัวเลขถูกหรือไม่กรณีที่ทายถูก จะให้ทายตัวต่อไป แต่ถ้าทายตัวเลขตัวแรกผิดแล้ว ก็ให้หยุดการทายตัวเลขทันที ดังนั้นกำหนดค่าเริ่มต้นของตัวแปร J มีค่าเท่ากับ 1 สำหรับตัวแปร Continue กำหนดค่าเริ่มต้นให้มีค่าเท่ากับ True
 - (5) เป็นคำสั่งทำงานวนรอบโดยค่า J ต้องมีค่าน้อยกว่าหรือเท่ากับ 4 และ Continue ต้องมีค่าเป็น True เท่านั้น ในกรณีนี้จะหยุดการทำงานวนรอบก็ต่อเมื่อ ทำงานวนรอบครบ 4 ครั้งแล้ว หรือทายตัวเลขแล้วผิด ซึ่งจะเปลี่ยนให้ Continue มีค่าเท่ากับ False นั่นเอง
 - (6) คำสั่งรับตัวเลขจำนวนเต็มทางแป้นพิมพ์เก็บในตัวแปร N
 - (7) เป็นคำสั่งในการตรวจสอบตัวเลขในเซตที่ตั้งไว้ โดยนำค่า N ตรวจสอบโดยใช้คำสั่ง IN ในกรณีที่ตัวเลขที่ทายอยู่ในเซตที่ตั้งไว้ จะเพิ่มค่า J ขึ้น 1 เพื่อวนรอบ (5) เพื่อทายตัวเลขตัวต่อไปในกรณีที่ทายผิดจะกระทำที่ (8)
 - (8) กรณีที่ทายตัวเลขผิด จะพิมพ์ข้อความว่า “คุณทายตัวเลขผิดแล้วค่ะ” และกำหนดให้ตัวแปร Continue มีค่าเท่ากับ False ต่อจากนั้นจะหลุดจากการทำงานวนรอบไปทำที่ (9)
 - (9) เป็นคำสั่งเงื่อนไขเพื่อตรวจสอบการทายตัวเลขที่ผู้ใช้ทายว่าเมื่อครบทั้ง 4 รอบของการทายตัวเลข ผู้ใช้ทายได้ถูกต้องหมดหรือไม่ ในกรณีที่ทายถูกต้อง กำหนดให้ Correct มีค่าเท่ากับ True มีผลให้โปรแกรมหลุดจากการทำงานวนรอบของ I
 - (10) กรณีทายเลขผิด จะมีข้อความบอกว่าการทายตัวเลขในรอบที่ I ผิดให้ทายใหม่โดยเพิ่มค่า I ขึ้น 1 และย้อนทำงานซ้ำที่ (3)

- (11) เมื่อทำงานจนครบสามรอบของการทนาย จะตรวจสอบว่าผู้ใช้โปรแกรมสามารถทนายได้อย่างถูกต้องหรือไม่ ถ้า Correct เป็น True แสดงว่าทนายถูก แต่ถ้าไม่ใช่ แสดงว่าทนายไม่ถูกนั่นเอง

แบบฝึกหัดท้ายบทที่ 13 ข้อ 2

จงเขียนโปรแกรมให้สำหรับหาว่าสีที่เลือกเป็นสีใดสีหนึ่งของธงชาติหรือไม่ โดยสีที่เลือกให้กำหนดจำนวนสีขึ้นมาเอง

Program Ex13_2;

Type Colour = (Red ,Blue ,Green ,White, Yellow, Pink,Brown,Violet,Orange);(1)

ITEM = (Red , White , Blue);(2)

Var Flag : Set of ITEM ;(3)

Choice : Colour;(4)

I : Integer;

Begin

Flag := [Red,Blue,Green,White, Yellow,Pink,Brown,Violet,Orange];(5)

WriteLn('โปรดเลือกสีที่ท่านต้องการตรวจสอบ');(6)

WriteLn('1. สีแดง 6. สีชมพู ');

WriteLn('2. สีน้ำเงิน 7. สีน้ำตาล');

WriteLn('3. สีเขียว 8. สีม่วง');

WriteLn('4. สีขาว 9. สีส้ม');

WriteLn('5. สีเหลือง ');

Write('เลือกสีใดสีหนึ่งต่อไปนี้?');readLn(I);(7)

Case I of(8)

1 : Choice := Red ;

2 : Choice := Blue;

3 : Choice := Green;

```

4 : Choice := White;
5 : Choice := Yellow;
6 : Choice := Pink;
7 : Choice := Brown;
8 : Choice := Violet;
9 : Choice := Orange;

End; {CASE..of}

If Choice IN Flag then Writeln('ท่านเลือกสีของธงชาติถูกต้อง') .....(9)
Else Writeln('ท่านเลือกสีของธงชาติไม่ถูกต้อง');

```

End.

คำอธิบายโปรแกรม

- (1) ประกาศ Colour เป็นชนิดข้อมูลที่ประกอบด้วยสีต่างๆ
- (2) ประกาศ ITEM เป็นชนิดข้อมูลที่ประกอบด้วยสีของธงชาติไทย
- (3) ประกาศตัวแปร Flag เป็นตัวแปรชนิดเซตของ ITEM
- (4) ประกาศตัวแปร Choice เป็นข้อมูลชนิด Colour
- (5) กำหนดข้อมูลเซตให้กับตัวแปร Flag
- (6) พิมพ์เมนูเลือกการทำงาน
- (7) ผู้ใช้เลือกสีจากเมนู เก็บในตัวแปร I
- (8) ในกรณีที่ผู้ใช้เลือกสีใด จะนำสีนั้นไปเก็บในตัวแปร Choice
- (9) นำสีที่ผู้ใช้เลือกตรวจสอบในสีของธงชาติ กรณีที่ใช่พิมพ์ "ท่านเลือกสีของธงชาติถูกต้อง"
ในกรณีที่เลือกผิด พิมพ์ "ท่านเลือกสีธงชาติไม่ถูกต้อง"

แบบฝึกหัดท้ายบทที่ 14

แบบฝึกหัดท้ายบทที่ 14 ข้อ 1

ให้สร้างข้อมูลชนิดที่มีการชี้ต่อของ 2 Linked list ซึ่งเก็บชื่อ และเงินเดือนของพนักงาน 2 บริษัท โดยข้อมูลที่เก็บใน Linked list นี้จะเรียงลำดับตามชื่อจากน้อยไปมาก

Program Ex14_1;

TYPE

String10 = String[10];

PersonPtr = ^Node;(1)

Node = record(2)

 Name : String10 ;

 Salary : real;

 next : PersonPtr ;

End;

VAR First , Second : PersonPtr;(3)

Procedure InsertLinkedList(VAR Head : PersonPtr ; N : String10 ; S : real);(4)

Var P , Prv , Pt : PersonPtr;(5)

Begin

 New(P);(6)

 P^.Name := N;

 P^.Salary := S ;

 P^.next := nil;

 Prv := nil;(7)

 Pt := head;

```

WHILE (Pt <> nil) and (Pt^.Name < N) do .....(8)
  Begin
    Prv := Pt;
    Pt := Pt^.next;
  End;

```

```

If Prv= nil then {insert first list} .....(9)
  Begin
    P^.next := Head ;
    Head := P ;
  End

```

```

Else
  Begin .....(10)
    P^.next := Pt ; {insert mid or last of list}
    Prv^.next := P ;
  End;

```

```

End;

```

```

Procedure Create(Var Head : PersonPtr); .....(11)

```

```

VAR N : String10;

```

```

S : real ;

```

```

Begin

```

```

  Head := nil; .....(12)

```

```

  Write('ชื่อพนักงาน');readln(N); .....(13)

```

```

  While N<>'*' do

```

```

    Begin

```

```

      Write('เงินเดือน'); readln(S); .....(14)
    End;
  End;

```

InsertLinkedlist(Head,N,S);(15)

Write('ชื่อพนักงาน');readln(N);(16)

End;

End;

Begin {MAIN}

Create(First);(17)

Create(Second);

End.

คำอธิบายโปรแกรม

- (1) ประกาศชนิดข้อมูล PersonPtr ให้ชี้ไปยัง Node
- (2) ประกาศ Node ข้อมูลประกอบด้วย 3 필ด์คือ 필ด์ Name ซึ่งเก็บชื่อ 필ด์ Salary เก็บเงินเดือน และฟิลด์ next ที่เป็นพอยเตอร์ที่ชี้ไปยัง Node ถัดไป
- (3) ประกาศตัวแปร First และ Second เป็นตัวแปรพอยเตอร์ 2 ตัวที่ชี้ไปยัง Node
- (4) เป็นโพรซีเจอร์ในการนำข้อมูล N และ S ในที่นี้คือชื่อและเงินเดือนของพนักงานเก็บในลิสต์เชื่อมโยงโดยที่มีตัวแปรพอยเตอร์ Head ซึ่งอ้างอิงกับตัวแปรพอยเตอร์ที่เรียกใช้ ซึ่งอยู่ที่ข้อมูลโหนดแรก
- (5) ประกาศตัวแปร P ,Pt ,Prv เป็นตัวแปรพอยเตอร์ที่ชี้ไปยังโหนด ตัวแปร P จะชี้ที่ข้อมูลโหนดใหม่ที่ต้องการแทรกเข้าไปในลิสต์ที่ Head ซึ่งอยู่การแทรกสามารถแทรกได้ 3 ลักษณะกล่าวคืออาจแทรกที่โหนดแรก แทรกระหว่างโหนด หรือแทรกเป็นโหนดสุดท้ายก็ได้ ดังนั้นเราต้องหาดำแหน่งในการแทรกเสียก่อน จึงกำหนดตัวแปรพอยเตอร์ขึ้นมาอีก 2 ตัวคือ Pt จะเป็นพอยเตอร์ที่ชี้ตำแหน่งหลังแทรก และ Prv เป็นพอยเตอร์ชี้ตำแหน่งก่อนแทรก
- (6) การทำงานภายในโพรซีเจอร์นี้เริ่มจากการขอเนื้อที่ว่าง 1 โหนดโดยให้ตัวแปรพอยเตอร์ P ชี้อยู่ต่อจากนั้นนำข้อมูลที่ส่งมาจากจุดเรียกใช้ เก็บในโหนดที่ P ชี้อยู่
- (7) กำหนดให้ Prv ชี้ที่ nil และ Pt ชี้ที่โหนดแรกของลิสต์

- (8) ทำงานวนรอบเพื่อหาตำแหน่งในการแทรกโดยเปรียบเทียบค่าของข้อมูลที่ต้องการแทรกกับข้อมูลที่เก็บในโหนดข้อมูลของลิสต์เริ่มจากโหนดแรก การวนรอบจะสิ้นสุดเมื่อ Pt เป็น nil หรือ Prv จะชี้อยู่ตำแหน่งก่อนแทรก และ Pt ชี้ตำแหน่งหลังแทรก
- (9) ในกรณีที่ Prv เป็น nil แสดงว่าโหนดใหม่จะแทรกหน้าลิสต์เก่า เราจะนำโหนดใหม่เชื่อมต่อกับลิสต์เก่า และเปลี่ยนให้ Head มาชี้ที่โหนดใหม่แทน
- (10) ในกรณีที่ Prv ไม่ใช่ nil แสดงว่าการแทรกอาจจะแทรกระหว่างโหนดหรือแทรกท้ายลิสต์ก็ได้ การเชื่อมโยงเพียงแต่นำโหนดก่อนหน้าเชื่อมโยงกับโหนดใหม่และนำโหนดใหม่เชื่อมโยงกับโหนดที่ตัวแปรพอยเตอร์ Pt ชี้อยู่
- (11) เป็นโพธิเจอรในการสร้างลิสต์ โดยส่งตัวแปรพอยเตอร์ที่ชี้ลิสต์มายังโพธิเจอร ซึ่งพารามิเตอร์ที่รับเป็นการอ้างอิงกัน การทำงานในโพธิเจอรจะส่งผลกับตัวแปรพอยเตอร์ที่ส่ง
- (12) กำหนดค่าเริ่มต้นให้แก่ตัวแปรพอยเตอร์มีค่าเท่ากับ nil
- (13) รับชื่อพนักงานคนแรกของลิสต์
- (14) กรณีที่ไม่หมดข้อมูล ในกรณีนี้ถ้าไม่ใช่ "*" จะรับข้อมูลพนักงานเพื่อเก็บในลิงค์ลิสต์เรื่อยๆ รับเงินเดือนของพนักงาน
- (15) เรียกใช้โพธิเจอรในนำข้อมูลที่รับเข้ามาแทรกในลิสต์โดยเรียงลำดับตามชื่อจากน้อยไปมาก ซึ่ง Head จะชี้ที่ข้อมูลในลิสต์โหนดแรก
- (16) อ่านชื่อพนักงานคนต่อไปทางแป้นพิมพ์ นำไปเปรียบเทียบกับที่ (14) ในกรณีเป็นคนสุดท้ายจะหยุดการทำงาน
- (17) เป็นโปรแกรมหลักที่มีการเรียกใช้โพธิเจอรเพื่อสร้าง 2 ลิงค์ลิสต์ เพื่อเก็บข้อมูลพนักงาน โดยลิสต์แรกมีตัวแปรพอยเตอร์ First ชี้อยู่ ลิสต์ที่สองมีตัวแปรพอยเตอร์ Second ชี้ ผลจากการทำงานจะได้พนักงานของสองบริษัทเชื่อมโยงใน 2 ลิงค์ลิสต์โดยมีการเรียงลำดับตามชื่อจากน้อยไปมาก

แบบฝึกหัดท้ายบทที่ 14 ข้อ 2

จงเขียนโปรแกรมหา Union และ Intersection ของ 2 linked list ที่เก็บเลขจำนวนเต็ม กำหนดให้ A และ B เป็นตัวแปรพอยเตอร์ที่ชี้ไปยังโครงสร้างข้อมูลที่เก็บเลขจำนวนเต็มและตัวชี้ต่อ ผลลัพธ์ของการทำงานจะได้เป็นสมาชิกทั้งหมดและสมาชิกที่ซ้ำกันของ 2 linked list นี้

Program EX14_2;

```
Type Ptr = ^Node; .....(1)
```

```
Node = record
```

```
    Data : integer;
```

```
    Next : Ptr;
```

```
End;
```

```
VAR A ,B : Ptr; .....(2)
```

```
Procedure Insert(VAR Head:Ptr); .....(3)
```

```
VAR P , Last : Ptr;
```

```
Begin
```

```
    NEW(P); .....(4)
```

```
    Readln(P^.Data);
```

```
    P^.Next := nil ;
```

```
    Last := P ; Head := P; .....(5)
```

```
    While P^.Data>0 do .....(6)
```

```
        Begin
```

```
            NEW(P);
```

```
            Readln(P^.Data);
```

```
            P^.Next := nil ;
```

```
            Last^.Next := P; .....(7)
```

```
            Last := P; .....(8)
```

```
        End;
```

```

End;
Procedure Union (A,B : Ptr); .....(9)
VAR  First , Second : Ptr;
      Found : Boolean;
Begin
  First := A; .....(10)
  While First<>nil do .....(11)
    Begin
      Write(First^.Data:5);
      First := First^.Next;
    End;
  Second := B; .....(12)
  While Second <> nil do .....(13)
    Begin
      First := A;
      Found := False;
      While (First<>nil) And (not Found) do .....(14)
        Begin
          If Second^.data = First^.data then .....(15)
            Found := True
          Else
            First := First^.next; .....(16)
          End;
          If not Found then Write(Second^.Data:5) .....(17)
          Else Second := Second^.Next;
        End;
      End;
    End;
  END;

```



```

Procedure Intersection(A,B : Ptr); .....(18)
Var Found : Boolean;
Begin
    While A<>nil do .....(19)
        Begin
            Found := False; .....(20)
            While B<>nil do .....(21)
                Begin
                    If A^.Data=B^.Data then Found := True .....(22)
                    Else B := B^.Next;
                End;
                If Found then Write(A^.Data:5) .....(23)
                Else A := A^.Next;
            End;
        End;
    End;
End;
Begin {MAIN}
    Insert(A); .....(24)
    Insert(B);
    Union(A,B); .....(25)
    Intersection(A,B); .....(26)
End.

```

คำอธิบายโปรแกรม

- (1) ประกาศโครงสร้างของลิสต์โดยกำหนดให้ Ptr เป็นชนิดของข้อมูลที่ชี้ไปยังโหนด
- (2) ประกาศตัวแปร A ,B เป็นชนิดพอยเตอร์ที่ชี้ไปยังโหนด
- (3) โพรซีเจอร์ในการนำข้อมูลจำนวนเต็มไปเก็บในลิสต์ที่มีตัวแปรพอยเตอร์ Head อ้างอิงอยู่ โดย Head จะชี้ที่ข้อมูลของลิสต์โหนดแรก

- (4) มีการประกาศตัวแปรพอยเตอร์ที่ใช้ในโพสิเจอร์ โดยประกาศให้ตัวแปร P เป็นตัวแปรพอยเตอร์ที่ชี้ไปยังโหนดข้อมูลใหม่ที่ต้องการเชื่อมต่อกับลิสต์ข้อมูล และตัวแปร last เป็นตัวแปรพอยเตอร์ที่ชี้ที่โหนดสุดท้ายของลิสต์ การทำงานเริ่มจากจองเนื้อที่ใหม่หนึ่งโหนด อ่านข้อมูลที่ต้องการจากแป้นพิมพ์นำมาเก็บในโหนดใหม่โดยให้ตัวแปรพอยเตอร์ P ชี้อยู่
- (5) กำหนดให้ Head และ Last ชี้ที่โหนดใหม่ที่ P ชี้อยู่
- (6) คำสั่งทำงานวนรอบในกรณีที่ยังมีข้อมูลต้องการเก็บในหน่วยความจำอยู่ เงื่อนไขในการทำงานคือกรณีที่ตัวเลขจำนวนเต็มมีค่ามากกว่า 0 จะทำงานวนรอบ(7)-(8) แต่ถ้าข้อมูลมีค่าน้อยกว่าหรือเท่ากับ 0 จะหยุดการทำงานวนรอบ การทำงานภายในเริ่มจากการจองเนื้อที่ว่างหนึ่งโหนด และรับข้อมูลมากเก็บในโหนดนี้ โดยมีตัวแปรพอยเตอร์ P ชี้อยู่
- (7) นำโหนดใหม่ที่ P ชี้อยู่มาเชื่อมต่อกับลิสต์โดยนำมาเชื่อมต่อหลังโหนดที่ Last ชี้อยู่
- (8) กำหนดให้ Last เลื่อนไปที่โหนดใหม่แทน มีผลให้ Last ชี้ที่โหนดสุดท้ายของลิสต์
- (9) โพสิเจอร์ Union วัตถุประสงค์เพื่อนำข้อมูลของทั้งสองลิสต์พิมพ์ออกมาทางจอภาพโดยมีการส่งตัวชี้โหนดแรกของลิสต์ 2 ลิสต์ที่ต้องการมา การทำงานมีการประกาศตัวแปรภายใน 3 ตัวด้วยกันคือ ตัวแปรพอยเตอร์ First และ Second และตัวแปร Found เป็นชนิดบูลีน
- (10) การทำงานเริ่มจากการพิมพ์ข้อมูลในลิสต์ที่ A ชี้อยู่ทั้งหมดออกมาทางจอภาพเสียก่อน โดยให้ First ชี้ที่โหนดแรกของลิสต์ที่ A ชี้อยู่
- (11) ใช้คำสั่งการทำงานวนรอบเพื่อเลื่อนให้ตัวชี้ First เลื่อนไปยังโหนดถัดๆไปที่โหนด โดยพิมพ์ข้อมูลที่บรรจุในแต่ละโหนดออกมาทางจอภาพ จนกระทั่ง First เลื่อนไปเป็น nil แสดงว่าหมดข้อมูลแล้ว จึงหยุดการทำงานวนรอบ
- (12) สำหรับอีกลิสต์ที่เหลือ กำหนดให้ Second ชี้ที่ลิสต์ที่พอยเตอร์ B ชี้อยู่
- (13) การทำงานจะทำงานวนรอบเพื่อเลื่อนโหนดของข้อมูลไปที่โหนดในการทำงานแต่ละรอบจนกระทั่งหมดข้อมูล
- (14) การทำงานภายใน Loop จะมี Loop ช้อนอีกชั้นหนึ่งเพื่อตรวจสอบว่ามีข้อมูลที่ซ้ำกันหรือไม่สำหรับข้อมูลในลิสต์ทั้งสอง กรณีที่ข้อมูลซ้ำกันจะไม่พิมพ์ เพราะได้พิมพ์ไปแล้ว จะ

- พิมพ์เฉพาะค่าของข้อมูลที่ไม่ซ้ำกันเท่านั้น จึงมีการกำหนดให้ตัวแปร Found มีค่าเป็น False
- (15) การทำงานจะมีการตรวจสอบการซ้ำกันของข้อมูลในกรณีที่ซ้ำจะกำหนดให้ Found มีค่าเป็น True และหลุดจาก Loop ตัวใน กรณีที่ไม่ซ้ำจะวนรอบจนกระทั่งหมดข้อมูลและหลุดจาก Loop เช่นกัน ไปที่(17)
 - (16) เลื่อนไปยังโหนดข้อมูลตัวถัดไป
 - (17) เมื่อหลุดจาก Loop ใน ตรวจสอบค่าของ Found ว่าเป็นอย่างไร กรณีที่ Found เป็น True แสดงว่าข้อมูลซ้ำจะเลื่อนโหนดของข้อมูล Loop นอกไปยังโหนดถัดไป แต่กรณีที่ Found เป็น False แสดงว่าข้อมูลไม่ซ้ำกัน จะพิมพ์ข้อมูลตัวที่ไม่ซ้ำออกทางจอภาพ
 - (18) โพรซีเจอร์ในการหาข้อมูลที่ซ้ำกันของทั้งสองลิสต์ โดยส่งตัวชี้โหนดข้อมูลแรกของลิสต์มายังโพรซีเจอร์นี้
 - (19) การทำงานจะคล้ายๆกับโพรซีเจอร์ที่แล้ว มีการทำงานวนสอง Loop โดย Loop นอกให้เลื่อนโหนดของลิสต์ข้อมูลที่ตัวแปรพอยเตอร์ A ซึ่งอยู่ที่โหนดจนกระทั่งหมดข้อมูล
 - (20) สำหรับ Loop ภายในเลื่อนโหนดของลิสต์ B ที่โหนดเช่นกันโดยมีตัวแปร Found ตรวจสอบการเหมือนกันของข้อมูลทั้งสองลิสต์ กำหนดเริ่มต้นให้มีค่าเท่ากับ False
 - (21) คำสั่งการทำงานวนรอบของ Loop ภายใน
 - (22) คำสั่งเงื่อนไขในการตรวจสอบข้อมูล กรณีที่เท่ากันจะกำหนดให้ Found มีค่าเท่ากับ true และหลุดจากการทำงานของ Loop ใน กรณีที่ไม่พบจะเลื่อนไปยังโหนดถัดไป
 - (23) กรณีที่หลุดจาก Loop ในแล้วจะตรวจสอบข้อมูลจาก Found กรณีที่ Found มีค่าเท่ากับ true แสดงว่าข้อมูลเหมือนกัน จะพิมพ์ข้อมูลนี้ออกทางจอภาพ ในกรณีที่ Found มีค่าเท่ากับ False แสดงว่าข้อมูลไม่เหมือนกันจะเลื่อนโหนดข้อมูลใน Loop นอกให้เลื่อนไปยังข้อมูลตัวต่อไป ทำงานวนรอบจนกระทั่งหมดข้อมูล จะได้ผลลัพธ์คือข้อมูลที่ซ้ำกันทั้งสองลิสต์นี้พิมพ์ออกทางจอภาพ
 - (24) เป็นการเรียกใช้โพรซีเจอร์ Insert จากโปรแกรมหลัก เพื่อสร้างลิสต์ข้อมูลในการเก็บเลขจำนวนเต็ม โดยเรียกครั้งแรกลิสต์ข้อมูลแรกมีตัวแปรพอยเตอร์ A ซึ่งอยู่ เรียกครั้งที่สองมีตัวแปรพอยเตอร์ B ซึ่งอยู่

- (25) เรียกใช้โพรซีเจอร์ Union ทำงานโดยส่งตัวแปรพอยเตอร์ A และ B ไปยังโพรซีเจอร์ การทำงานจะนำข้อมูลทั้งสองลิสต์พิมพ์ออกทางจอภาพเฉพาะที่ไม่ซ้ำกัน
- (26) เรียกใช้โพรซีเจอร์ Intersection ทำงานโดยส่งตัวแปรพอยเตอร์ A และ B ไปยังโพรซีเจอร์ การทำงานจะนำข้อมูลที่ซ้ำกันพิมพ์ทางจอภาพ

แบบฝึกหัดท้ายบทที่ 14 ข้อ 3

จงเขียนโปรแกรมนับจำนวน โหนดทั้งหมดที่มีอยู่ใน Linked list หนึ่ง

Function Count (Head:Ptr): Integer;

VAR N : integer;

Begin

 N:=0 ;

 While Head<>nil do

 Begin

 N := N+1;

 Head := Head^.Next;

 End;

End;

คำอธิบาย

การทำงานของโปรแกรมเราสามารถสร้างได้โดยเรียกโพรซีเจอร์ Insert จากแบบฝึกหัดที่ 14 ข้อที่ 2 ได้ และสามารถเรียกใช้ได้เหมือนกัน สำหรับการนับจำนวนโหนด สามารถเขียนเป็นฟังก์ชันโดยสามารถเรียกใช้ได้ดังนี้

 N := Count(A);

โดยที่ N เป็นตัวแปรจำนวนเต็มซึ่งหมายถึงจำนวนโหนดข้อมูลที่ตัวแปรพอยเตอร์ A ชี้ที่ข้อมูลโหนดแรกนั่นเอง