

Unit 4

ประเภทของข้อมูลและตัวดำเนินการ Data Types and Operator

จากตัวอย่างโปรแกรมอย่างง่ายที่ผ่านมาจะเห็นว่าถ้าจะให้คอมพิวเตอร์ประมวลผลจะต้องทำการเขียนชุดคำสั่งที่เรียกว่าโปรแกรมให้กับคอมพิวเตอร์ และการทำงานของโปรแกรมอาจต้องมีการรับข้อมูลเข้าไปเพื่อทำการประมวลผล ข้อมูลที่รับเข้าไปอาจอยู่ในรูปของค่าคงที่ หรือตัวแปร โดยค่าคงที่จะเป็นค่าที่มีค่าคงที่ตลอดโปรแกรม ส่วนตัวแปรจะเป็นค่าในหน่วยความจำที่สามารถเปลี่ยนแปลงได้ในการทำโปรแกรม ตัวอย่างในโปรแกรมที่ผ่านมาจะเห็นว่าการประกาศตัวแปรเป็น integer เพื่อเก็บค่าที่เป็นเลขจำนวนเต็ม และค่าในตัวแปรนั้นสามารถนำมาประมวลผลและมีการเปลี่ยนแปลงได้

4.1 ประเภทของข้อมูล

การประกาศข้อมูลในการเขียนโปรแกรมจะเป็นการกำหนดชื่อชนิดของข้อมูล หรือกำหนดประเภทของข้อมูลขึ้นมาใหม่ โดยข้อมูลในภาษาปาสคาลอาจแบ่งออกได้เป็น 4 กลุ่มดังต่อไปนี้

- ข้อมูลชนิดซิมเปิล (simple type)
- ข้อมูลประเภทสตริง (string type)
- ข้อมูลประเภทโครงสร้าง (structure type)
- ข้อมูลประเภทพอยน์เตอร์ (pointer type)

ในหัวข้อนี้จะกล่าวถึงข้อมูลชนิดซิมเปิลและข้อมูลชนิดสตริงก่อน ส่วนข้อมูลประเภทอื่น ๆ จะกล่าวต่อไปในภายหลัง

4.1.1 ข้อมูลชนิดซิมเปิล

ข้อมูลชนิดซิมเปิลแบ่งได้เป็นข้อมูลประเภทลำดับ (ordinal type) และข้อมูลประเภทจำนวนจริง (Real Data Type) โดยข้อมูลแบบลำดับเป็นข้อมูลที่มีค่าเป็นลำดับแน่นอน เช่นตัวเลขที่ใช้ในการนับ ลำดับตัวอักษรเป็นต้น ในภาษาปาสคาลยังแบ่งข้อมูลชนิดลำดับออกได้หลายประเภท ในที่นี้จะกล่าวถึงข้อมูลประเภทจำนวนเต็ม ข้อมูลอักขระ และข้อมูลตรรก

ข้อมูลชนิดจำนวนเต็ม (Integer Data Type)

ข้อมูลประเภทนี้จะใช้เก็บตัวเลขที่เป็นจำนวนเต็ม ในคอมพิวเตอร์จะใช้หน่วยความจำในการเก็บข้อมูลถ้าหากคอมพิวเตอร์ใช้หน่วยความจำ 8 บิตหรือ 1 ไบต์ในการเก็บข้อมูลจะทำให้เก็บข้อมูลที่เป็นเลขฐานสิบได้ในช่วง 0 ถึง 255 ข้อมูลชนิดจำนวนเต็มนี้ยังแบ่งได้หลายประเภทขึ้นกับขนาดของหน่วยความจำที่คอมพิวเตอร์ใช้เก็บ โดยข้อมูลประเภทต่างแสดงได้ดังตารางต่อไปนี้

ประเภท	ช่วงของข้อมูลที่เก็บได้	ขนาดหน่วยความจำ
byte	0 .. 255	1 ไบต์
shortint	-128 .. 127	1 ไบต์
integer	-32768 .. 32767	2 ไบต์
word	0 .. 65535	2 ไบต์
longint	-2,147,483,648 .. 2,147,483,647	4 ไบต์

ตารางแสดงประเภทของข้อมูลและช่วงของข้อมูลที่เก็บได้

ข้อมูลประเภทตัวอักขระ (Character Data Type)

ข้อมูลประเภทนี้จะเป็นตัวอักขระหนึ่งตัว ซึ่งเป็นไปตามตารางรหัส ASCII ประกอบด้วยข้อมูลที่เป็นตัวอักษร ตัวเลข และอักขระพิเศษ ข้อมูลประเภทนี้จะเก็บข้อมูลแบบลำดับได้ เนื่องจากเรียงตามลำดับรหัส ASCII ข้อมูลประเภทนี้จะใช้เนื้อที่ในการเก็บหนึ่งไบต์ การประกาศข้อมูลให้เป็นตัวอักขระจะใช้คำว่า Char

ข้อมูลประเภทตรรก (Boolean Data Type)

จะเป็นค่าทางลอจิก ได้แก่จริง(True) กับเท็จ (False) จะใช้ในคำสั่งควบคุมเพื่อตัดสินใจการทำงาน ในการเรียงลำดับจะให้ค่าที่เป็นเท็จมีลำดับก่อนค่าที่เป็นจริง

ข้อมูลประเภทจำนวนจริง (Real Data Type)

ข้อมูลประเภทนี้จะเป็นจำนวนจริงหรือเลขทศนิยม ข้อมูลประเภทนี้จะจัดลำดับก่อนหลังได้ยาก จึงไม่เป็นข้อมูลชนิดลำดับเนื่องจากทศนิยมมีได้หลายตำแหน่ง ข้อมูลจำนวนจริงนี้ยังแบ่งออกได้หลายประเภท โดยแต่ละประเภทจะใช้หน่วยความจำในการเก็บแตกต่างกัน ทำให้เก็บข้อมูลได้ต่างกัน ดังตารางต่อไปนี้

ประเภท	ช่วงของข้อมูลที่เก็บได้	ขนาดหน่วยความจำ
Real	2.9 E -39 .. 1.7 E +38	6 ไบต์
Single	1.5 E -45 .. 3.4 E +38	4 ไบต์
Double	5.0 E -324 .. 1.7 E +308	8 ไบต์
Extended	1.9 E -4951 .. 1.1 E +4932	10 ไบต์
Comp	$-2^{63} + 1 .. 2^{63} - 1$	8 ไบต์

ตารางแสดงประเภทของข้อมูลชนิดจำนวนจริง

หมายเหตุ 2.9 E -39 มีค่าเท่ากับ 2.9×10^{-39}

4.1.2 ข้อมูลประเภทสตริง (string type)

ข้อมูลประเภทนี้จะเป็นการนำตัวอักษรมาต่อเรียงกับเป็นข้อความ โดยสามารถเก็บตัวอักษรได้ 255 ตัว

4.2 ส่วนประกาศ (Declaration Part)

จากโครงสร้างของโปรแกรมในบทที่ผ่านมาจะเห็นว่าในโปรแกรมจะมีส่วนประกาศ ซึ่งอยู่ต่อจากส่วนชื่อโปรแกรม โดยส่วนประกาศนี้ในโปรแกรมบางโปรแกรมอาจไม่มี แต่ถ้าหากโปรแกรมต้องการใช้ค่าคงที่ หรือใช้ตัวแปรต่าง ๆ ผู้เขียนโปรแกรมจะต้องมีส่วนนี้สำหรับประกาศชนิดของข้อมูล (Type) ลาเบล (Label) ค่าคงที่ (Const) ตัวแปร (Var) โฟซีเยอร์ (Procedure) และฟังก์ชัน (Function)

4.2.1 การประกาศค่าคงที่

ค่าคงที่เป็นค่าในหน่วยความจำที่มีค่าคงที่ตลอดโปรแกรม ในการประกาศค่าคงที่จะเป็นการกำหนดชื่อให้ค่าคงที่ ถ้าในโปรแกรมส่วนใดเรียกชื่อที่ประกาศไว้ก็จะได้ข้อมูลตามที่กำหนด การประกาศค่าคงที่ทำได้สองลักษณะดังต่อไปนี้

```

CONT      ชื่อ      =      ค่าคงที่;
CONT      ชื่อ      =      ประเภทของข้อมูล = ค่าคงที่

```

ในการประกาศค่าคงที่นี้เราอาจประกาศเพียงตัวเดียว หรือประกาศพร้อมกันหลายตัวก็ได้ ตัวอย่างเช่น

```

CONT      size      =      5;
          Perfect    =      100.0;
          Pi         =      3.14159;

```

การประกาศค่าคงที่ในลักษณะนี้ ปาสคาลจะเลือกประเภทของข้อมูลที่เหมาะสมให้กับชื่อค่าคงที่เอง โดย size จะเป็นข้อมูลประเภท Integer ส่วน Prefect และ Pi จะเป็นข้อมูลประเภท Real แต่ในการประกาศค่าคงที่บางครั้งเราสามารถกำหนดประเภทของข้อมูลลงไปด้วยได้ ตัวอย่างเช่น

```

CONT      Min       =      integer = 0;

```

โดยจะให้ชื่อ Min มีค่าเท่ากับ 0 โดยเป็นเลขจำนวนเต็ม ในการประกาศค่าคงที่นี้ข้อมูลที่เก็บจะอยู่ในหน่วยความจำ ปาสคาลจะมองชื่อค่าคงที่เป็นหน่วยความจำตำแหน่งหนึ่งที่เก็บค่านั้น

4.2.2 การประกาศตัวแปร

ในภาษาปาสคาลสามารถประกาศตัวแปรได้ โดยชื่อตัวแปรจะเป็นตำแหน่งหน่วยความจำที่เก็บข้อมูลอยู่ โดยเป็นข้อมูลตามประเภทที่กำหนดและข้อมูลที่เป็นตัวแปรนี้จะสามารถเปลี่ยนแปลงได้โดยการทำงานของโปรแกรม ในการเขียนโปรแกรมคอมพิวเตอร์ส่วนใหญ่จะต้องมีการประกาศตัวแปรเสมอ ตัวอย่างเช่นในโปรแกรมในบทที่ผ่านมาที่มีการรับข้อมูลจากแป้นพิมพ์เข้าสู่คอมพิวเตอร์ โดยจะมีการประกาศตัวแปรเอาไว้ ข้อมูลที่รับเข้ามาจะถูกนำไปเก็บในตัวแปรที่ประกาศเอาไว้ การประกาศตัวแปรสามารถทำได้ดังนี้

```

VAR <ชื่อตัวแปร,...> :      ประเภทของข้อมูล
    <ชื่อตัวแปร,...> :      ประเภทของข้อมูล

```

ในการประกาศตัวแปรสามารถประกาศครั้งละหลายตัวได้ ถ้าหากเป็นตัวแปรประเภทเดียวกันจะใช้เครื่องหมาย , คั่น ตัวอย่างเช่นถ้าหากจะประกาศตัวแปรชื่อ Data1 และ Data2 สำหรับเก็บจำนวนเต็มสามารถทำได้ดังนี้

```

VAR Data1,Data2 :      integer;

```

แต่ถ้าถ้าประกาศตัวแปรให้ชื่อ Data1 เก็บเลขจำนวนเต็ม และ Data2 เก็บเลขจำนวนจริงสามารถทำได้ดังนี้

```
VAR Data1 : integer;
    Data2 : Real;
```

ถ้าหากประกาศตัวแปรเป็น

```
VAR CH : Char;
```

หมายความว่าให้ตัวแปร CH เป็นตัวแปรสำหรับเก็บอักขระตัวเดียว เราสามารถให้ตัวแปรนี้เก็บรหัสแอสกีของตัวอักขระได้ เช่นถ้าต้องการเก็บอักขระที่มีรหัสแอสกีเป็น 201 สามารถทำได้ดังนี้

```
CH := #201;
```

ต่อมาถ้ามีการเขียนคำสั่งเป็น

```
WRITELN(CH);
```

จะเป็นการพิมพ์ตัวอักขระที่มีรหัสแอสกีเท่ากับ 201

ตัวอย่าง โปรแกรมต่อไปเป็นตัวอย่งโปรแกรมที่มีการประกาศค่าคงที่ และตัวแปร โดยโปรแกรมนี้จะคำนวณค่าภาษี 7 % และแสดงทางหน้าจอ

```
PROGRAM Tax;
CONST
    Taxrate = 0.07;
VAR
    itemcost,salestax : real;
BEGIN
    WRITE('Please enter cost of item: ');
    READLN(itemcost);
    Salestax := taxrate * itemcost;
    WRITELN('Item cost is ',itemcost);
    WRITELN('Sales tax is ',salestax);
END.
```

ในการประกาศประเภทของข้อมูลให้กับตัวแปร เมื่อมีการนำตัวแปรนั้นมาใช้งานข้อมูลที่เก็บในตัวแปรจะต้องอยู่ภายในขอบเขตช่วงของตัวแปรที่กำหนดด้วย ตัวอย่างเช่นถ้าเขียนสเตตเมนต์ดังต่อไปนี้

```

VAR
    I,J,K : Integer;
BEGIN
    I := 20000;
    J := 20000;
    K := I + J;
    Writeln(K);
END

```

จากสแตตเมนต์ข้างต้นตัวแปร K ควรจะมีค่าเท่ากับ 40000 แต่ผลลัพธ์ที่พิมพ์ออกมาจะไม่ถูกต้อง เนื่องจากตัวแปร K ถูกประกาศเป็น Integer ซึ่งจะเก็บข้อมูลได้ในช่วง -32,768 ถึง 32,767 จากตัวอย่างสแตตเมนต์ดังกล่าวควรจะประกาศตัวแปร K เป็น LongInt

4.2.3 การประกาศประเภทของข้อมูลใหม่

การกำหนดข้อมูลชนิดใหม่นอกจากข้อมูลที่มีอยู่แล้วจะใช้คำว่า TYPE โดยข้อมูลที่กำหนดอาจเป็นช่วงของข้อมูลหรือลำดับของข้อมูลก็ได้ โดยมีรูปแบบดังนี้

```

TYPE
    ชื่อข้อมูลใหม่ = ข้อมูลที่กำหนด;
    ชื่อข้อมูลใหม่ = ข้อมูลที่กำหนด;
    .....
    .....

```

อย่างเช่นถ้าหากต้องการประกาศชื่อข้อมูลใหม่ให้ชื่อว่า Days โดยประกอบด้วยคำย่อของวัน 7 วันเช่น SUN MON TUE WED THU FRI และ SAT ตัวอย่างของการประกาศ TYPE เช่น

```

TYPE
    Days = (SUN,MON,TUE,WED,THU,FRI,SAT);
    Scores = 0..100;
    WeekDays = MON..FRI;

```

จากการประกาศประเภทของข้อมูลข้างต้นจะทำให้ Days เป็นข้อมูลประเภทใหม่ที่ประกอบด้วย SUM เรียงลำดับไปถึง SAT โดยมี SUM อยู่ลำดับที่ 0 และ SAT อยู่ลำดับที่ 6 ส่วนข้อมูลประเภท Scores จะเป็นข้อมูลที่อยู่ในช่วง 0 ถึง 100 เท่านั้น ส่วน WeekDays เป็นข้อมูลช่วงย่อยที่อยู่ระหว่าง MON ถึง FRI ตามชนิดของข้อมูลที่กำหนดโดย Days


4.3 การตั้งชื่อ

ในการเขียนโปรแกรมจะต้องมีการกำหนดชื่อต่าง ๆ หรือไอดีเฟนติไฟเออร์ (Identifier) เพื่อให้โปรแกรมทำงาน ตัวอย่างเช่นการกำหนดชื่อโปรแกรม ชื่อของตัวแปรต่าง ๆ เป็นต้น ในตัวอย่างที่ผ่านมาเราได้เห็นการตั้งชื่อโปรแกรมและตัวแปรมาบ้างแล้ว การตั้งชื่อในภาษาปาสคาลมีรูปแบบดังนี้

1. ชื่อจะต้องไม่ซ้ำกับคำสงวน (Reserved word) และคำมาตรฐานที่คอมไพเลอร์รู้จัก
2. จะต้องขึ้นต้นด้วยตัวอักษร (A-Z,a-z)
3. ตัวต่อไปต้องเป็นตัวอักษรหรือตัวเลขหรือสัญลักษณ์ _
4. การตั้งชื่อจะต้องไม่มีช่องว่าง

คำสงวน เป็นคำที่มีความหมายที่โปรแกรมรู้จัก โดยมีรูปแบบการใช้งานที่แน่นอน ส่วนคำมาตรฐานเป็นคำที่มีความหมายอยู่แล้ว โปรแกรมสามารถเรียกใช้งานได้เลย คำสงวนในภาษาปาสคาลได้แก่

ABSOLUTE	EXTERNAL	MOD	SHR
AND	FILE	NIL	STRING
ARRAY	FOR	NOT	THEN
BEGIN	FORWARD	OBJECT	TO
CASE	FUNCTION	OF	TYPE
CONST	GOTO	OR	UNIT
CONSTRUCTOR	IF	PACKED	UNTIL
DESTRUCTOR	IMPLEMENTATION	PROCEDURE	USES
DIV	IN	PROGRAM	VAR
DO	INLINE	RECORD	VIRTUAL
DOWNTO	INTERFACE	REPEAT	WHILE
ELSE	INTERRUPT	SET	WITH
END	LABEL	SHL	XOR

 **คำถาม** การตั้งชื่อต่อไปนี้อันใดใช้ได้ อันใดใช้ไม่ได้

1. Average
2. 1234A
3. A1234
4. 1234
5. A1234A
6. A1234
7. Hot Dog

44 ภาษาปาสคาล

- | | |
|----------------|-----------|
| 3. Program | 8. \$XYZ |
| 4. A*B | 9. George |
| 5. 506-74-3981 | 10. 4x2 |

4.4 ตัวดำเนินการ (Operator)

ในการเขียนโปรแกรมตัวดำเนินการจะเป็นตัวที่ทำหน้าที่รวมค่าต่าง ๆ และกระทำกับค่าต่าง ๆ ให้เป็นค่าเดียวกัน อย่างเช่นในโปรแกรมในบทที่ผ่านมาที่มีการนำข้อมูลที่เป็นตัวแปรมาคูณกับค่าคงที่ ซึ่งจะต้องใช้ตัวดำเนินการทางคณิตศาสตร์เพื่อทำการคูณ ตัวดำเนินการมีหลายประเภทดังต่อไปนี้

4.4.1 ตัวดำเนินการเลขคณิต

ใช้สำหรับทำการคำนวณทางคณิตศาสตร์ เช่นบวก ลบ คูณ หาร โดยจะนำข้อมูลตัวหนึ่งไปกระทำกับอีกตัวหนึ่ง โดยให้ผลลัพธ์เป็นตัวเลขทางคณิตศาสตร์ ตัวดำเนินการทางคณิตศาสตร์แบ่งออกได้ดังต่อไปนี้

ตัวดำเนินการ	กระบวนการ	ข้อมูลที่ถูกกระทำ	ข้อมูลผลลัพธ์
+	บวก (Addition)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
-	ลบ (Subtraction)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
*	คูณ (Multiplication)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
/	หาร (real number Division)	จำนวนจริง	จำนวนจริง
DIV	หารดิบ (integer number Division)	จำนวนเต็ม	จำนวนเต็ม
MOD	หารมอด (Modula)	จำนวนเต็ม	จำนวนเต็ม

ตารางแสดงตัวดำเนินการเลขคณิต

ตัวอย่าง ตัวอย่างต่อไปเป็นการใช้ตัวดำเนินการบวกลบ และคูณ

$$\begin{aligned}3 + 4 &= 7 \\7.0 - 3.0 &= 4.0 \\6 * 1.5 &= 9.0 \\1.5 - 1 &= 0.5 \\2.25 * 1.5 &= 3.375 \\5.8 + 3 &= 8.8\end{aligned}$$

จากตัวอย่างจะเห็นว่าถ้านำเลขจำนวนจริงไปกระทำกับเลขใด ผลลัพธ์ที่ออกมาจะเป็นเลขจำนวนจริง

ตัวอย่าง ตัวอย่างต่อไปเป็นการใช้ตัวดำเนินการหารแบบต่าง ๆ

$9 / 2$	=	4.5
$9 \text{ div } 2$	=	4
$9 \text{ mod } 2$	=	1
$18 \text{ mod } 2$	=	0
$-14 \text{ div } 3$	=	-4
$-14 \text{ mod } 3$	=	-2
$14 \text{ mod } -3$	=	2
$18.2 / 2$	=	9.1
$18.2 \text{ div } 2$	=	error

จากตัวอย่างจะเห็นว่าเลขจำนวนเต็มหารเลขจำนวนเต็มผลลัพธ์ที่ได้จะเป็นเลขจำนวนจริง เลขจำนวนจริงถูกหารด้วยเลขจำนวนเต็มจะได้เลขจำนวนจำนวนจริง แต่ถ้าเลขจำนวนจริงถูกหารด้วยเลขจำนวนเต็มจะหารไม่ได้

ในสแตตเมนต์ของการทำงานต่าง ๆ อาจมีตัวดำเนินการหลายตัวได้ โดยผลลัพธ์ที่ได้จะเกิดการกระทำของตัวดำเนินการแต่ละตัว ดังตารางต่อไปนี้

การกระทำ	ผลลัพธ์
$12 \text{ DIV } 5 * 3$	6 จำนวนเต็ม
$6 * 5 / 10 * 2 + 10$	16.0 จำนวนจริง
$(6 * 5) / (10 * 2) + 10$	11.5 จำนวนจริง
$(6 * 5) / (10 * 2 + 10)$	1.0 จำนวนจริง
$(6 * 5) / (10 * (2 + 10))$	0.25 จำนวนจริง

จากตารางจะเห็นว่าถ้ามีตัวดำเนินการหลายตัว ผลลัพธ์จะเกิดจากการกระทำของตัวดำเนินการแต่ละตัว ถ้าหากมีการใช้วงเล็บการกระทำใด ๆ จะกระทำในวงเล็บก่อน

ตัวอย่าง

$$5 \text{ MOD } 2 + 14 \text{ DIV } 3 - 6$$

$$1 + 4 - 6 \quad : \text{ ทำ MOD และ DIV ก่อนทำการบวก}$$

$$5 - 6$$

$$-1$$

ตัวอย่าง

$$3 * (4 \text{ MOD } (6 \text{ DIV } 2)) + 5$$

$$3 * (4 \text{ MOD } 3) + 5 \quad ; \text{ทำในวงเล็บในก่อนทำวงเล็บนอก}$$

$$3 * 1 + 5$$

$$8$$

ตัวอย่าง

$$7.5 - 10 / 4 * (2 + 3)$$

$$7.5 - 10 / 4 * 5$$


$$7.5 - 2.5 * 5$$

$$7.5 - 12.5$$


$$-5.0$$

 คำถาม หลังจากคอมไพเลอร์ทำคำสั่งต่อไปนี้ เอาต์พุตที่ได้จะเป็นอย่างไร

- ก. 26 MOD 4 ข. 26 DIV 4 ค. 26 / 4

 คำถาม ในการทำคำสั่งต่อไปนี้ ผลลัพธ์ที่ได้จะเป็นอย่างไร

- ก. 1 + 2 * 3 + 4 ข. 6 + 4 / 2 + 3 ค. 2 / 3 * 4

 คำถาม พิจารณาโปรแกรมดังนี้ $2 + 3 * 4$ จะมีค่าเท่ากับ 20 เพราะ $(2 + 3) * 4$ หรือเท่ากับ 14 เพราะ $2 + (3 * 4)$

4.4.2 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการเปรียบเทียบ (Relation Operators) จำนำข้อมูลสองค่ามาเปรียบเทียบกัน โดยข้อมูลทั้งสองค่าจะต้องเป็นข้อมูลประเภทเดียวกัน ผลลัพธ์ที่ได้จะเป็นค่าจริงหรือเท็จ

ตัวดำเนินการ	กระบวนการ
=	เท่ากับ
<>	ไม่เท่ากับ
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
>	มากกว่า
<	น้อยกว่า

ตัวดำเนินการเปรียบเทียบ

4.4.3 ตัวดำเนินการทางตรรก (Logical Operator)

ตัวดำเนินการทางตรรก (logical operator) เมื่อกระทำกับค่าใดผลลัพธ์ที่ออกมาจะเป็นจริงหรือเท็จ ตัวดำเนินการทางตรรกแสดงได้ดังตารางต่อไปนี้

ตัวดำเนินการ	การกระทำ
AND	AND ค่าสองค่า ถ้าค่าทั้งสองเป็นจริงผลลัพธ์จะเป็นจริง
OR	OR ค่าสองค่า ถ้าค่าทั้งสองเป็นเท็จผลลัพธ์จะเป็นเท็จ
XOR	XOR ค่าสองค่า ถ้าค่าทั้งสองต่างกันผลลัพธ์จะเป็นจริง
NOT	เปลี่ยนค่าจากจริงเป็นเท็จ จากเท็จเป็นจริง

ตัวอย่าง การกระทำต่อไปนี้จะให้ผลลัพธ์เป็นจริง

$$(5 = 4 + 1) \text{ AND } (18 \leq 6 * 4)$$

ถ้าหากในประโยคภาษาปาสคาลมีการใช้ตัวดำเนินการหลายตัว ภาษาปาสคาลจะจัดลำดับความสำคัญการทำงานก่อนหลังดังต่อไปนี้

ตัวดำเนินการ	ลำดับการทำงาน
NOT	1
*, /, DIV, MOD, AND	2
+, -, OR, XOR	3
>, >=, <, <=, <>, =	4

ตัวอย่าง การกระทำต่อไปนี้ผลลัพธ์ที่ได้จะเป็นจริงหรือเท็จ ถ้า X = 6 , Y = 13 และ Z = 4.2

$$((X \lt Y \text{ DIV } 2) \text{ OR } (Z \leq Y) \text{ AND } (\text{NOT } (Z = X / 2)))$$

วิธีทำ

$$((6 \lt 13 \text{ DIV } 2) \text{ OR } (4.2 \leq 13) \text{ AND } (\text{NOT } (4.2 = 6 / 2)))$$

$$((6 \lt 6) \text{ OR } (4.2 \leq 13) \text{ AND } (\text{NOT } (4.2 = 6 / 2)))$$

$$(\text{false}) \text{ OR true) AND true}$$

$$\text{true}$$

ดังนั้นผลลัพธ์ที่ได้จะเป็นจริง

4.4.4 ตัวดำเนินการระดับบิต (Bitwise Operator)

ตัวดำเนินการประเภทนี้จะกระทำกับข้อมูลที่เป็นเลขจำนวนเต็ม และผลลัพธ์ที่ได้จะเป็นเลขจำนวนเต็ม โดยจะดำเนินการทางตรรกแบบบิตต่อบิต ตัวดำเนินการที่ใช้ได้แก่ shl (เลื่อนบิตข้อมูลไปทางซ้าย), shr (เลื่อนบิตข้อมูลไปทางขวา) ,and,or,xor และ not ในการใช้ตัวดำเนินการประเภทนี้ ให้มองตัวเลขจำนวนเต็มในรูปของเลขฐานสองดังตัวอย่างโปรแกรมต่อไปนี้
โปรแกรมที่ 4. ตัวอย่างตัวดำเนินการแบบบิต AND

```
PROGRAM AndOperator;
USES CRT;
VAR Byte1,Byte2,Byte3 : Byte;
BEGIN
    CLRSCR;
    Byte1 := 77;
    Byte2 := 62;
    Byte3 := Byte1 AND Byte2;
    WRITELN(Byte3);
END.
```

เมื่อโปรแกรมทำงานให้มองข้อมูลแต่ละตัวในตัวแปรในรูปเลขฐานสอง และนำข้อมูลที่ได้จากการดำเนินการเก็บในตัวแปร Byte3 โดยมีขั้นตอนการทำงานดังนี้

ตำแหน่งบิตของข้อมูล	7	6	5	4	3	2	1	0
Byte1	0	1	0	0	1	1	0	1
AND								
Byte2	0	0	1	1	1	1	1	0
ผลลัพธ์ Byte3	0	0	0	0	1	1	0	0

ตัวอย่างโปรแกรมที่ 4.1 ตัวอย่างตัวดำเนินการแบบบิต OR

```
PROGRAM OrOperator;
USES CRT;
VAR Byte1,Byte2,Byte3 : Byte;
BEGIN
    CLRSCR;
    Byte1 := 77;
    Byte2 := 62;
    Byte3 := Byte1 OR Byte2;
    WRITELN(Byte3);
END.
```

ตำแหน่งบิตของข้อมูล	7	6	5	4	3	2	1	0
Byte1	0	1	0	0	1	1	0	1
OR								
Byte2	0	0	1	1	1	1	1	0
ผลลัพธ์ Byte3	0	1	1	1	1	1	1	1

ตัวอย่างโปรแกรมที่ 4.2 ตัวอย่างตัวดำเนินการแบบบิต XOR

```
PROGRAM XorOperator;
USES CRT;
VAR Byte1,Byte2,Byte3 : Byte;
BEGIN
    CLRSCR;
    Byte1 := 77;
    Byte2 := 62;
    Byte3 := Byte1 XOR Byte2;
    WRITELN(Byte3);
END.
```

50 ภาษาปาสคาล

ตำแหน่งบิตของข้อมูล	7	6	5	4	3	2	1	0
Byte1	0	1	0	0	1	1	0	1
XOR								
Byte2	0	0	1	1	1	1	1	0
ผลลัพธ์ Byte3	0	1	1	1	0	0	0	1

พิจารณาโปรแกรมต่อไปนี้

<pre>PROGRAM Drill; VAR x, y : real; BEGIN x := 426.5; y := 0.5; WRITELN('x is ',x); WRITELN('y is ',y) END.</pre>	<p>เอาต์พุต</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>x is 4.2650000000E+02 y is 5.0000000000E-01</pre> </div>
--	---

หลังจากรันโปรแกรมจะเห็นว่าเอาต์พุตจากการพิมพ์เลขจำนวนจริงที่ได้จะมีเทอม E+02 หมายความว่าค่า 4.265 จะต้องคูณด้วย 10^2 ซึ่งมีค่าเท่ากับ 426.5 ในการใช้คำสั่งพิมพ์ WRITELN เราควรระบุเอาต์พุตที่ได้จะให้แสดงผลที่ตำแหน่ง มีทศนิยมกี่ตำแหน่ง ในการกำหนดรูปแบบเอาต์พุตของคำสั่ง WRITELN จะเป็นดังนี้

WRITELN (RealExpr : width : p);

โดยที่ RealExpr เป็นเลขจำนวนจริง และ p เป็นจำนวนทศนิยม ส่วน width เป็นความกว้างของเอาต์พุต ถ้าหากระบุเป็น 0 เอาต์พุตที่ได้จะมีความกว้างน้อยที่สุดที่สามารถจะแสดงข้อมูลนั้นได้ ส่วนถ้าระบุจำนวนทศนิยมมากกว่าค่าที่มีอยู่จริง โปรแกรมปาสคาลจะเติม 0 ไปด้านหลัง พิจารณาตัวอย่างต่อไปนี้

X := 78.291	เอาต์พุต
WRITELN(X : 0 : 1);	78.3
WRITELN(X : 0 : 2);	78.29
WRITELN(X : 0 : 0);	78
WRITELN('X = ', X : 0 : 3);	X = 78.291
WRITELN(X : 0 : 4);	78.2910

ในการเขียนโปรแกรมและให้ประมวลผลทางคณิตศาสตร์ เราสามารถให้ตัวเลขจำนวนเต็มมากระทำกับเลขจำนวนจริงได้ โดยผลลัพธ์ที่ได้จะเป็นเลขจำนวนจริง ตัวอย่างโปรแกรมต่อไปเป็นโปรแกรมที่รับค่าข้อมูลในหน่วยของนิ้วและแปลงข้อมูลให้เป็นเซนติเมตร โดยประกาศตัวแปรรับค่านิ้วเป็นจำนวนเต็ม จากนั้นนำไปแปลงเป็นเซนติเมตรโดยคูณกับค่า 2.54 แสดงผลทางหน้าจอ

```
PROGRAM    Inchtocm;
VAR        inches : integer;
           centimeters : real;
BEGIN
    WRITE ('enter number of inches ');
    readln(inches);
    centimeters := 2.54 * inches;
    write (inches, ' inches equals ');
    writeln (centimeters : 0 : 1, ' centimeters')
END.
```

ผลลัพธ์จากการรันโปรแกรมจะเป็นดังนี้

```
enter number of inches 4
4 inches equals 10.2 centimeters
```

ในการเขียนโปรแกรมหากมีการใช้ตัวแปรจะต้องทราบว่าตัวแปรประเภทนั้นเก็บข้อมูลได้ในช่วงใด ตัวอย่างเช่นถ้าหากประกาศตัวแปรเป็น integer และหลังจากประมวลผลตัวแปรนั้นมีค่าเกิน 32,767 จะทำให้ค่านั้นมีค่าเป็นลบ แต่ถ้าหากประกาศตัวแปรขนาดใหญ่เกินไปจะทำให้โปรแกรมนั้นใช้หน่วยความจำมาก ถ้าหากต้องการให้คอมพิวเตอร์คำนวณ $2,000 * 2,000$ ซึ่งผลลัพธ์ที่ได้จะมีค่าเป็น 4 ล้าน ตัวแปรที่ประกาศควรจะเป็น longint โดยอาจเขียนคำสั่งได้ดังนี้

```
VAR    profit : longint;
.....
.....
profit := 2000 * 2000;
writeln ('profit $ ', profit);
```

เอาต์พุต

```
profit $ 4000000
```

โปรแกรมข้างบนนี้ถ้าหากประกาศตัวแปรเป็น integer โปรแกรมจะทำงานผิดพลาด แต่ถ้าหากตัวเลขที่ได้จากการคำนวณมีค่าไม่เกิน 32,767 ควรประกาศตัวแปรเป็น integer เพื่อจะใช้หน่วยความจำน้อยกว่า นอกจากนี้การประกาศตัวแปรขนาดไม่เท่ากันเราสามารถนำมาประมวลผลรวมกันได้ อย่างเช่นถ้าตัวแปร m เป็น integer และตัวแปร n เป็น longint และให้คอมพิวเตอร์ประมวลผล

$$p := m + n;$$

ตัวแปร p จะต้องประกาศเป็น longint ในการประมวลผลโปรแกรมถ้าหากนำตัวแปรขนาดต่างกันมาประมวลผลรวมกัน ค่าที่ได้จะเป็นตัวแปรชนิดที่มีขนาดใหญ่กว่า การกระทำแบบนี้เรียกว่า common type

4.5 ข้อผิดพลาดของโปรแกรม

ในการเขียนโปรแกรมและคอมไพล์อาจมีข้อผิดพลาดขึ้นได้ ถ้าหากคอมพิวเตอร์พบข้อผิดพลาดต่าง ๆ มันจะไม่สามารถทำโปรแกรมต่อไปได้ โดยทั่วไปแล้วข้อผิดพลาดของโปรแกรมมีสามลักษณะคือ 1. Syntax 2. Run-time 3. Logical

Syntax Error ข้อผิดพลาดประเภทนี้เกิดจากการเขียนโปรแกรมไม่เป็นไปตามข้อกำหนด เช่นขาดเครื่องหมายไปบางประเภท เขียนโปรแกรมที่มีรูปแบบไม่ตรงตามข้อกำหนด ประกาศตัวแปรผิดพลาด โดยจะมีการแจ้งข้อผิดพลาดตอนคอมไพล์โปรแกรม พิจารณาตัวอย่างโปรแกรมต่อไปนี้

```
PROGRAM Drill;
VAR n : interger;           1
BEGIN
    WRITELN('Hello ')      2
    WRITELN('How are you); 3
    n := 2;
    WRITELN('n eqalls ',n)
END.
```


ในการเขียนโปรแกรมข้างต้นจะเกิด syntax error ขึ้น เนื่องจากเขียนโปรแกรมไม่ถูกต้องตามข้อกำหนด โดยข้อผิดพลาดที่ 1 จะเห็นว่าสัปดาห์ integer ผิด ข้อผิดพลาดที่ 2 จะขาดเครื่องหมาย semicolon (;) ข้อผิดพลาดที่ 3 จะขาดเครื่องหมาย quote mark โดยทั่วไปแล้วเมื่อเกิดข้อผิดพลาดประเภทนี้ คอมไพเลอร์จะแจ้งข้อผิดพลาดและแสดงเคอร์เซอร์ที่ตำแหน่งนั้น เมื่อแก้ไขให้ถูกต้องจะทำให้คอมไพเลอร์ต่อไปได้

Run-Time Error เป็นข้อผิดพลาดที่เกิดขึ้นตอนรันโปรแกรม บางครั้งจะเรียกข้อผิดพลาดนี้ว่า execution error ตัวอย่างเช่นถ้ามีการเขียนโปรแกรมดังต่อไปนี้

```
C := 0;
```

```
Y := X / C;
```

เมื่อโปรแกรมทำงานจะเกิดการหารด้วยค่าศูนย์ขึ้น โดยโปรแกรมจะแจ้งข้อความ division by zero ออกมาทางจอภาพ

Logical Error ข้อผิดพลาดแบบนี้จะเกิดจากการออกแบบโปรแกรม โดยจะไม่มีแจ้งข้อผิดพลาดออกมาทางคอมไพเตอร์ ตัวอย่างเช่นถ้าหากเราต้องการหาค่าเฉลี่ยของการบวกกันของตัวแปร B กับ C และเขียนโปรแกรมให้ทำงานเป็น

```
AVG := B + C / 2;
```

ถ้าหากค่าใน B เป็น 60 และค่าใน C เป็น 80 ผลที่ได้จะมีค่าเท่ากับ 100 ซึ่งคอมไพเตอร์คำนวณได้ถูกต้อง แต่การเขียนโปรแกรมนั้นถูกออกแบบมาผิด ถ้าหากต้องการให้คอมไพเตอร์หาค่าเฉลี่ย ในกรณีนี้จะต้องให้ B บวกกับ C อยู่ในวงเล็บ

สำหรับกรณีที่ใช้ประเภทของตัวแปรไม่ถูกต้อง เช่นประกาศตัวแปรเป็น Integer และให้ตัวแปรนั้นเก็บค่าที่มากกว่า 32767 ก็เป็นข้อผิดพลาดประเภทนี้เช่นกัน

แบบฝึกหัด

1. จงเขียนเลขต่อไปนี้เป็นเลขทศนิยมในระบบเลขฐานสิบทั่วไป
 - 1.1 $1.437E+03$
 - 1.2 $5.462E-01$
 - 1.3 $5.462E-03$
 - 1.4 $1.437E+05$
2. ถ้าหากตัวแปร x เป็นประเภทจำนวนจริง และเก็บค่า 824.176 จงบอกเอาต์พุตเมื่อมีการทำคำสั่งต่อไปนี้

```
WRITELN(X : 0 : 1);
WRITELN(X : 0 : 2);
WRITELN(X);
```

3. จงหาผลลัพธ์จากการทำต่อไปนี้
 - 3.1 $8 * 6 / 3 * 4$
 - 3.2 $8 * 6 / (3 * 4)$
 - 3.3 $(8 * 6) / 3 * 4$
 - 3.4 $1 + 4 * 6 + 8 / 4 + 4$
4. ถ้าหาก $a = 49$, $b = 5$ และ $C = 3$ จงบอกผลลัพธ์จากการทำงานต่อไปนี้
 - 4.1 $a \text{ MOD } b * c + 1$
 - 4.2 $24 / c * 4$
 - 4.3 $7 + 2 \text{ div } c - 1$
 - 4.4 $a \text{ MOD } (b * c) + 1$
 - 4.5 $a \text{ DIV } b \text{ MOD } 2$
 - 4.6 $48 / (c * 2) * 4$
5. จงหาที่ผิดในโปรแกรมต่อไปนี้

```
PROGRAM Drill;
VAR x,y : integer;
BEGIN
  x := 6
  y := x / 2;
  WRITELN(x, ' ',y)
END.
```

```
PROGRAM Drill;
VAR n, total : integer;
    avg : real;
BEGIN
  n := 0;
  total := 50;
  avg := total / n;
  WRITELN(avg : 0 : 1)
END.
```