บทที่ 5

ข้อมูลอักขระ สตริง และรูปแบบการแสดงผล
Char and String and Formatting Output

ในบทนี้จะกล่าวถึงรายละเอียดของตัวแปรสองชนิดคืออักขระ(char) สตริง(string) โดยข้อมูลแบบอักขระจะเป็นตัวอักขระที่มีอยู่ในตารางรหัส ASCII ส่วนข้อมูลแบบสตริงจะประกอบด้วยตัวอักขระตั้งแต่ 0 ตัวขึ้นไปมาต่อเรียงกัน สตริงที่มีอักขระจำนวนศูนย์ตัวเรียกว่าเอ็มตี้สตริง (Empty String) หรือนัลสตริง (Null String) ข้อมูลแบบอักขระและแบบสตริงเราสามารถกำหนดให้มีความกว้างในการแสดงผลได้ตามที่เราต้องการ

5.1 ตัวแปรประเภทอักขระ

ตัวแปรประเภทอักขระ (Char) จะเก็บตัวอักษร หรือตัวเลข หรือเครื่องหมายต่าง ๆ จำนวนหนึ่งตัว สามารถใช้คำสั่ง WRITELN แสดงผลทางเอาต์พุต และใช้คำสั่ง READLN รับค่าทางอินพุตได้ พิจารณาตัวอย่างโปรแกรมต่อไป จะประกาศตัวแปรชื่อ grade เป็นตัวแปรประเภทอักขระ

```
PROGRAM lettergrade;
VAR grade : char;
BEGIN
        WRITELN('Enter your grade ');
        READLN(grade);
        WRITELN('You received the grade of ',grade)
END.
```

เมื่อรันโปรแกรมเครื่องจะให้เราใสเกรดเข้าไป โดยสามารถใส่ตัวอักขระได้หนึ่งตัว อย่างเช่นถ้าใส่ B ในตัวแปร grade จะเก็บค่า 'B' หรือเก็บรหัส ASCII ของ B โดยเอาต์พุตจากการรันโปรแกรมจะเป็นดังนี้

Enter your grade

B

You received the grade of B

5.2 ตัวแปรประเภทสตริง

โดยทั่วไปแล้วตัวแปรประเภทสตริงจะใช้เก็บข้อมูลที่นำมาแสดงผล ไม่นิยมนำไปคำนวณ การใส่ค่าตัวแปรสตริงจะต้องอยู่ภายในเครื่องหมายฝนทอง (‘) ปิดเปิด ถ้าหากตัวแปรใดถูกกำหนดให้เป็นสตริง ตัวแปรนั้นจะสามารถเก็บข้อความที่มีความยาวได้ไม่เกิน 255 ตัวอักษร ตัวอย่างเช่นถ้าหากประกาศตัวแปรเป็น

```
var  LastName  :  String;
```

จากนั้นใส่ข้อมูลเป็น

```
LastName  :=  'TWAT';
```

คอมพิวเตอร์จะนำค่า TWAT ไปใส่ในตัวแปร LastName โดยตัวแปรนี้จะถูกจองเนื้อที่หน่วยความจำไป 255 ไบต์ แต่จะถูกเก็บข้อมูลเพียง 4 ไบต์

โปรแกรมต่อไปจะเป็นการประกาศตัวแปร initial เป็น char และประกาศตัวแปร LastName เป็น string เมื่อรันโปรแกรมจะให้ป้อนข้อมูลเข้าไป และแสดงผลทางหน้าจอ

```
PROGRAM  name;
VAR      initial      :  char;
         LastName  :  string;
BEGIN
         WRITE('Enter first initial:  ');
         READLN(initial);
         WRITE('Enter your last name:  ');
         READLN(LastName);
         WRITELN(LastName,   ' , ' ,initial, ' . ')
END.
```

ผลจากการรันโปรแกรมจะเป็นดังนี้

Enter first initial : M

Enter your last name : TWAT

TWAT , M.

ในการประกาศตัวแปรเป็นสตริงถ้าหากไม่ระบุค่าว่าจะเก็บจำนวนกี่ตัว เครื่องจะกำหนดให้เป็น 255 ตัวซึ่งจะใช้หน่วยความจำ 255 ไบต์ แต่การประกาศตัวแปรแบบนี้ถ้าหากข้อมูลมีน้อยจะทำให้เสียเนื้อที่หน่วยความจำ โดยทั่วไปแล้วการประกาศตัวแปรประเภทสตริงจะมีการระบุขนาดของข้อมูลที่เก็บ ตัวอย่างเช่น

```
VAR name : string[10];
```

ทำให้ตัวแปร name เก็บค่าสตริงไม่เกิน 10 ตัว ถ้าหากตัวแปร name มีค่าดังนี้

```
name := 'John Doe'
```

จะใช้หน่วยความจำทั้งหมด 8 ตำแหน่ง และเหลือสองตำแหน่ง

J	o	h	n		D	o	e		

แต่ถ้าหากตัวแปร name มีค่าเป็น

```
name := 'Weatherspoon'
```

โปรแกรมจะทำงานผิดพลาด เนื่องจากข้อมูล name มีทั้งหมด 12 ตัว การเก็บในหน่วยความจำนั้นจะตัดสองตัวสุดท้ายทิ้งไป

ในการเขียนโปรแกรมด้วย Turbo Pascal นั้นจะมีฟังก์ชั่นสำหรับหาความยาวสตริงคือ ฟังก์ชั่น length โดยค่าที่ให้ออกมาจะเป็นเลขจำนวนเต็ม ตัวอย่างการใช้งานเป็นดังโปรแกรมต่อไปนี้

```
PROGRAM How_long;
VAR   message : string[100];
BEGIN
        message := 'So Long';
        WRITELN(length(message))
END.
```

เมื่อรันโปรแกรมเอาต์พุตที่ได้จะเป็น 7 โดยการนับอักขระจะนับที่ว่างด้วย

5.3 การกำหนดรูปแบบการแสดงผล

ในการเขียนโปรแกรมให้เอาต์พุตที่ได้แสดงผลในรูปของตาราง เราจะต้องกำหนดความกว้างของอักขระที่จะแสดงผล โดยข้อมูลที่แสดงผลอาจเป็นข้อมูลสตริง จำนวนเต็ม หรือจำนวนจริงก็ได้ การกำหนดรูปแบบแสดงผลในคำสั่ง WRITELN ทำได้ดังนี้

WRITELN (StrExpr : width)

ถ้าหากข้อมูล StrExpr เป็นข้อมูลสตริงที่จะแสดงผล ตามด้วยเครื่องหมาย : ส่วน width เป็นความกว้างของข้อมูลที่จะพิมพ์ โดยเริ่มนับจากทางขวาสุดของข้อมูล ตัวอย่างต่อไปเป็นการกำหนดรูปแบบการแสดงผล โดยตัวแปร name มีค่าเป็น 'Jones'

```
WRITELN ('ABCDEFGH');
WRITELN ('NOW' : 6);
WRITELN (name : 6);
```

เอาต์พุตที่ได้จากการทำคำสั่งจะเป็น

```
ABCDEFGH
   NOW
 Jones
```

จากเอาต์พุตที่ได้จะเห็นว่า การแสดงคำว่า NOW ตัว W จะตรงกับตัว F และเหลือที่ว่างไว้ด้านหน้า 3 คอลัมพ์ เนื่องจากคำสั่ง WRITELN จะให้แสดงสตริงที่มีความกว้างเป็น 6 สำหรับการกำหนดรูปแบบการแสดงผลที่ข้อมูลที่แสดงผลเป็นตัวเลขจำนวนเต็มนั้น ความกว้างของข้อมูลก็จะเริ่มนั้บจากทางซ้ายสุดเช่นกัน

โปรแกรมต่อไปแสดงการเปรียบเทียบของการกำหนดรูปแบบแสดงผลเลขจำนวนเต็มและไม่กำหนดรูปแบบแสดงผล โดยตัวแปร a ,b , c และ d เป็นตัวแปรประเภทจำนวนเต็ม โดย a = 14 , b = 3254 , c = 8 และ d = 95 ให้สังเกตการแสดงผลของชุดคำสั่งทั้งสอง

```
WRITELN ('table');
WRITELN (a);
WRITELN (b);
WRITELN (c);
WRITELN (d);
```

```
WRITELN ('tables');
WRITELN (a : 5);
WRITELN (b : 5);
WRITELN (c : 5);
WRITELN (d : 5);
```

เอาต์พุตที่ได้จะเป็นดังต่อไปนี้

```
table
14
3254
8
95
```

```
table
  14
 3254
    8
   95
```

สำหรับการกำหนดรูปแบบการแสดงผลที่เป็นจำนวนจริงเราได้เคยศึกษามาบ้างแล้ว โดยมีรูปแบบเป็น

WRITELN (RealExpr : width : p)

โดยที่ RealExpr เป็นเลขจำนวนจริง width เป็นความกว้าง ส่วน p เป็นจำนวนตำแหน่งของทศนิยม ในการแสดงผลเป็นเลขจำนวนจริงนี้ การระบบรูปแบบการแสดงผลจะมีความสำคัญมาก เพราะถ้าหากไม่ระบุเครื่องจะแสดงผลเป็นตัวเลขทางวิทยาศาสตร์ ซึ่งจะมีหลายหลัก

ตัวอย่าง ถ้าหากตัวแปร x และ y เป็นตัวแปรประเภท real และ x = 29.431 และ y = 57.128 เมื่อคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
WRITELN ('ABCDEFGH');
WRITELN (X : 6 : 2);
WRITELN (Y : 6 : 2);
```

เอาต์พุตที่ได้จะเป็น

```
ABCDEFGH
 29 . 43
 57 . 13
```

ถ้าหากกำหนดรูปแบบเอาต์พุตน้อยกว่าความกว้างที่ถูกต้อง เครื่องจะแสดงค่าที่ถูกต้องออกมา ตัวอย่างเช่น

```
WRITELN (547:2);
WRITELN (89.463 : 1 : 2);
```

```
547
89.46
```

5.4 ฟังก์ชั่นทางคณิตศาสตร์

ที่ผ่านมาเราได้ศึกษาการใช้งานฟังก์ชั่น length มาแล้ว โดยฟังก์ชั่นนี้จะคืนค่าความยาวของสตริงมาให้ การเรียกใช้ฟังก์ชั่นต่าง ๆ ที่มีอยู่ในโปรแกรมแล้วเรียกว่า built – in function ในการใช้ฟังก์ชั่นจะต้องใส่ค่าเข้าไปในฟังก์ชั่น ที่เรียกว่าอาร์กิวแมนต์ (argument) ในเทอร์โบปาสคาลยังมีฟังก์ชั่นทางคณิตศาสตร์ให้ใช้อีกมากมาย ตัวอย่างเช่นถ้าหากใช้ฟังก์ชั่นในการหาค่ารากที่สอง ถ้าใส่ค่าอาร์กิวแมนต์เป็น 9 ฟังก์ชั่นนี้จะคืนค่า 3 ออกมา ฟังก์ชั่นต่าง ๆ ที่น่าสนใจมีดังนี้

ฟังก์ชั่น SQRT และ SQR

ฟังก์ชั่น sqrt จะคืนค่ารากที่สองของค่าตัวเลขบวกออกมา โดยค่าที่คือจะเป็นข้อมูลประเภท Real ส่วนค่าอาร์กิวแมนต์จะเป็น Integer หรือ Real ก็ได้

ฟังก์ชั่น sqr จะตรงข้ามกับฟังก์ชั่น sqrt โดยจะคือค่ายกกำลังสองของค่าอาร์กิวแมนต์ โดยค่าอาร์กิวแมนต์จะเป็นข้อมูลประเภท Integer หรือ Real ก็ได้ ส่วนค่าที่คือออกมาจะเป็นค่าประเภทเดียกกับอาร์กิวแมนต์

ตัวอย่าง เมื่อคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
WRITELN (SQRT (2) : 5 : 3);
WRITELN (SQRT (9) );
WRITELN (SQR (9));
```

ผลลัพธ์ที่ได้จะเป็น

```
1.414
 3.0000000000E+00
81
```

จะเห็นว่าในการแสดงผลของค่ารากของ 9 จะแสดงเป็นเลขทางวิทยาศาสตร์ (scientific notation) เราสามารถจัดรูปแบบการแสดงได้โดยใช้คำสั่ง

WRITELN (SQRT (9) : 1 : 0):

<u>คำถาม</u> โปรแกรมต่อไปนี้เป็นตัวอย่างการหาด้านตรงข้ามมุมฉากของสามเหลี่ยมมุมฉาก โดยให้ป้อนด้าน a และ ด้าน b เข้าไป อยากทราบว่าในที่ว่างที่กำหนดควรเขียนโปรแกรมอย่างไร

$$hypot^2 = a^2 + b^2$$

```
PROGRAM Pythagorean;
VAR     a , b , hypot  :  real;
BEGIN
        WRITE ('Enter the lengths of the two legs  ');
        READLN (a , b);
        hypot  := ……..;
        WRITELN ('Hypotenuse  is  ',hypot : 6 : 3)
END.
```

คำตอบ ในที่ว่างอาจเขียนโปรแกรมได้ดังนี้

```
hypot  :=  sqrt (a * a  +  b * b);          หรือ
hypot  :=  sqrt  (sqr(a) + sqr(b));
```

ฟังก์ชั่นทางคณิตศาสตร์อื่น ๆ ที่หน้าสนใจเป็นดังตารางต่อไปนี้

ฟังก์ชั่น	การคำนวณ	ชนิดอาร์กิวแมนต์	ชนิดของผลลัพธ์
abs(x)	ค่าสัมบูรณ์ของ x	integer หรือ real	เหมือนอาร์กิวแมนต์
* arctan(x)	ค่า Arc tangent ของ x	integer หรือ real	Real
* cos(x)	ค่า Cosine ของ x	integer หรือ real	Real
exp(x)	ค่า e^x เมื่อ e = 2.718…..	integer หรือ real	Real
ln(x)	ค่า Natural log ของ x	integer หรือ real	Real
pi	ค่า π		Real
random	สุ่มค่าข้อมูล		
round(x)	หาค่าจำนวนเต็มของ x	real	Integer
* sin(x)	หาค่า Sine ของ x	integer หรือ real	Real
sqr(x)	หาค่า กำลังสองของ x	integer หรือ real	เหมือนอาร์กิวแมนต์
sqrt(x)	หารากที่สองของ x	integer หรือ real	Real
trunc(x)	ตัดทศนิยมทิ้ง	real	Integer

* ค่าอาร์กิวเมนต์เป็นเรเดียน

การใช้ฟังก์ชั่นทางคณิตศาสตร์นั้นจะต้องใส่ชนิดของอาร์กิวแมนต์ให้ถูกต้อง อย่างเช่น ถ้าหากต้องการใช้ฟังก์ชั่นทางตรีโกน ค่าอาร์กิวแมนต์จะต้องเป็นค่าของเรเดียน ถ้าต้องการหาค่า sine ของมุม 30 องศา เราจะต้องแปลงค่ามุมให้เป็นค่าเรเดียนเสียก่อนจึงจะทำการคำนวณได้ โดยอาจทำการคำนวณด้วย deg * (π/180) ตัวอย่างเช่น

```
WRITE ('Enter angle in degrees  ');
READLN (deg);
radians  :=  deg * pi / 180;
WRITE ('Sine of  ',deg, ' degrees  = ',sin(radians) : 6 : 3);
```

ตัวอย่างการใช้ฟังก์ชั่นต่าง ๆ แสดงได้ดังนี้

การกระทำ	ผลลัพธ์
round(3.8)	4
trunc(3.8)	3
round(6.2)	6
round(8 / 3)	3
sqrt (abs(-16))	4.000
trunc (sqrt(15.9))	3
sin (pi/6)	0.5

5.5 ฟังก์ชั่นทางอักขระ

ฟังก์ชั่นภายในของเทอร์โบปาสคาลที่กระทำกับข้อมูลประเภทอักขระหรือ char ได้แก่

- ORD คืนค่ารหัส ASCII ตามตัวอักษรที่ระบุ
- CHR คืนค่าตัวอักษรตามรหัส ASCII ที่ระบุ
- UPCASE เปลี่ยนตัวอักษรตัวเล็กเป็นตัวใหญ่
- PRED คืนค่าที่มีลำดับก่อนค่านี้
- SUCC คืนค่าที่มีลำดับหลังค่านี้

ตัวอักขระทุกตัวที่มีอยู่ในแป้นพิมพ์รวมทั้งรหัสควบคุม คอมพิวเตอร์จะรับรู้ได้โดยดูจากรหัส ASCII ประจำตัวของมัน คำว่า ASCII ย่อมาจาก American Standard Code for Information Interchange โดยระบบคอมพิวเตอร์ถ้าต้องการรับค่าข้อมูลต่าง ๆ ที่เป็นตัวอักขระ

จะต้องผ่านการเข้ารหัสเป็นรหัส ASCII อย่างเช่น ตัว R จะแทนด้วย 82 ตัว r จะแทนด้วย 144 ดังตารางรหัส ASCII ท้ายเล่ม

ฟังก์ชั่น ORD และ CHR จะทำงานตรงข้ามกัน โดยฟังก์ชั่น ORD จะรับค่าอาร์กิวแมนต์ที่เป็นตัวอักขระ และคืนค่าเป็นรหัส ASCII ของอักขระตัวนั้น ตัวอย่างเช่น ORD('R') ค่าที่คืนออกมาจะเป็น 82 ส่วนฟังก์ชั่น CHR จะรับค่าอาร์กิวเมนต์ที่เป็นรหัส ASCII และจะคืนค่าออกมาเป็นตัวอักขระตามที่ระบุตามรหัส ASCII นั้น ตัวอย่างเช่น CHR (82) ค่าที่คือออกมาจะเป็นตัว R

ตัวอย่าง พิจารณาคำสั่งและผลลัพธ์จากการทำคำสั่งต่อไปนี้

```
WRITELN (ORD('B'));
WRITELN (ORD('>'));
WRITELN (CHR(78));
WRITELN (CHR(63));
WRITELN (ORD(CHR(89)));
```

ผลลัพธ์ที่ได้จากการทำคำสั่งจะเป็น

```
66
62
N
?
89
```

รหัส ASCII Bell ซึ่งมีหมายเลขเป็น 7 จะไม่แสดงผลทางจอภาพ แต่จะส่งเสียงออกทางลำโพงแทน ตัวอย่างเช่นถ้าหากต้องการให้ลำโพงมีเสียงจะเขียนคำสั่งได้ดังนี้

```
WRITELN (CHR(7));
```

ถ้าหากในโปรแกรมมีการประกาศประเภทของข้อมูลดังต่อไปนี้

```
TYPE month  =  (Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec);
```

จากที่ทราบมาแล้วในหัวข้อที่ 4.2.3 ว่าการประกาศข้อมูลใหม่ด้วย TYPE จะทำให้ข้อมูลถูกเรียงเป็นลำดับได้ และข้อมูลนี้สามารถใช้ฟังก์ชั่น ORD ได้เช่นกัน โดยฟังก์ชั่นจะคืนค่าลำดับออกมาเช่น

```
ORD(Feb)      คืนค่า   1
ORD(Dec)      คืนค่า   11
ORD(Jun)      คืนค่า   5
ORD(Jan)      คืนค่า   0
```

ฟังก์ชั่นเหล่านี้สามารถนำมาสร้างเป็นนิพจน์ได้เช่น

ORD(‘5’) – ORD(‘0’) = 53 – 48 = 5

ORD(‘6’) – ORD(‘0’) = 54 – 48 = 6

CHR(0 + ORD(‘0’)) = ‘0’

CHR(1 + ORD(‘0’)) = ‘1’

ฟังก์ชั่น UPCASE จะรับค่าอาร์กิวแมนต์ของตัวอักษร และคือค่ามาเป็นตัวอักษรตัวใหญ่ ดังตัวอย่างคำสั่งต่อไปนี้

UPCASE(‘g’) คืนค่า ‘G’

UPCASE(‘G’) คืนค่า ‘G’

UPCASE(‘4’) คืนค่า ‘4’

สำหรับฟังก์ชั่น PRED และ SUCC ตัวอย่างการใช้งานเช่น

PRED(‘C’) คืนค่า ‘B’

PRED(‘8’) คืนค่า ‘7’

PRED(3) คืนค่า 2

SUCC(‘a’) คืนค่า ‘b’

SUCC(0) คืนค่า 1

SUCC(-3) คืนค่า -2

ถ้าหากในโปรแกรมมีการประกาศประเภทของข้อมูลใหม่ดังนี้

TYPE days = (Sun,Mon,Tue,Wed,Thu,Fri,Sat); ซึ่งจะเป็นข้อมูลแบบลำดับ สามารถใช้ฟังก์ชั่น PRED และ SUCC ได้เช่นกัน

PRED(Sat) คืนค่า Fri

PRED(Wed) คืนค่า Tue

PRED(Mon) คืนค่า Sun

SUCC(Sun) คืนค่า Mon

ในการเขียนโปรแกรมคอมพิวเตอร์บางครั้งจะมีการนำตัวอักขระหรือสตริงมาเปรียบเทียบกัน โดยใช้ตัวดำเนินการเปรียบเทียบ เช่น > หรือ >= การเปรียบเทียบนี้คอมพิวเตอร์จะนำรหัส ASCII ของอักขระมาเปรียบเทียบกัน ว่าค่าใดก่อนหลัง ตัวอย่างเช่นถ้าคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
if (symbol  >= ‘A’) and (symbol  <= ‘M’)
   then    writeln (‘Yes’)
   else    writeln (‘No’);
```

ถ้าหากตัวแปร symbol มีค่าเป็นตัวอักขระต่าง ๆ เอาต์พุตที่ได้จะเป็นดังนี้

symbol = ‘g’, symbol =‘G’, symbol = ‘R’, symbol = ‘r’ กรณีของ symbol = ‘G’ เท่านั้นที่เอาต์พุตจะเป็น Yes เนื่องจากรหัส ASCII ระหว่าง ‘A’ กับ ‘M’ จะอยู่ในช่วง 65 ถึง 77

สำหรับในการเปรียบเทียบสตริงมักใช้ตัวดำเนินการเปรียบเทียบ 6 ตัวคือ <,<=,>,>=,=, <> มาเปรียบเทียบสตริงสองตัว โดยการเปรียบเทียบคอมพิวเตอร์จะนำรหัส ASCII ของอักขระแต่ละตัวมาเปรียบเทียบกัน โดยเริ่มจากอักขระตัวแรก แต่ถ้าหากอักขระตัวแรกของสตริงทั้งสองเป็นตัวเดียวกัน คอมพิวเตอร์จะเปรียบเทียบอักขระตัวต่อไป ตัวอย่างเช่น

ABC < BBC	เป็นจริง เพราะรหัส ASCII ของ A น้อยกว่า B
ACB > ABC	เป็นจริง เพราะรหัส ASCII ของ C มากกว่า B

ข้อควรระวัง

1. ถ้าหากตัวแปร word1 มีค่าเป็น ‘baker’ และตัวแปร word2 มีค่าเป็น ‘Charlie’ เมื่อคอมพิวเตอร์ทำคำสั่ง if word1 < word2 ผลลัพธ์ที่ได้จะเป็นเท็จ เนื่องจากรหัส ASCII ของ b มีค่ามากกว่าของ C ซึ่งนักเขียนโปรแกรมบางคนจะเข้าใจว่า b มากกว่า C
2. สตริงที่มาเปรียบเทียบ ถ้าเป็นตัวเลขจำนวนหลักจะต้องเท่ากัน ตัวอย่างเช่น ‘99’ > ‘200’ เนื่องจากว่า 9 มีรหัส ASCII มากกว่า 2

5.6 ตัวดำเนินการ +

ตัวดำเนินการ + นอกจากจะใช้กับระบบตัวเลขแล้วยังสามารถใช้กับตัวอักขระและสตริงได้อีกด้วย ตัวอย่างเช่นถ้าตัวแปร first,last และ name เป็นตัวแปรประเภท string และคอมพิวเตอร์ทำงานต่อไปนี้

```
first := ‘John’
last := ‘Bull ‘;
name := first + ‘   ‘ + last;
writeln (name);
```

จะเป็นการนำค่าที่อยู่ใน first มารวมกับ last โดยมีที่ว่างรวมอยู่ด้วย เอาต์พุตที่ได้จะเป็น

John Bull

ตัวอย่าง ถ้าหากตัวแปร card เป็นตัวแปรประเภทสตริง และคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
WRITELN  ('7'  +  chr (3) );
card  :=  'Q' + chr (4) );
WRITELN (card);
```

เอาต์พุตที่ได้จะเป็น

7♥

Q◊

แบบฝึกหัด

1. จงบอกเอาต์พุตเมื่อคอมพิวเตอร์ทำคำสั่งต่อไปนี้

ก.

```
m := 431;
n  := 57;
WRITELN ('ABCDEFG');
WRITELN (m: 6);
WRITELN (n: 6);
WRITELN (m:4);
WRITELN (m: 2);
```

ข.

```
x := 3.8412;
y := 47.162;
WRITELN ('ABCDEFG');
WRITELN (x : 6: 1);
WRITELN (y : 6: 1);
WRITELN (x : 6: 3);
WRITELN (y : 3: 2);
```

2. ถ้าหาก k = 675 , l = 42 , m = 18 และ n = 5 จงบอกเอาต์พุตเมื่อคอมพิวเตอร์ทำคำสั่งต่อไปนี้

```
WRITELN ('ABCDEFGH');
WRITE (k : 4);
WRITE (l : 4);
WRITELN;
WRITE (m : 4);
WRITE (n : 4);
```

3. จงบอกเอาต์พุตจากการทำคำสั่งต่อไปนี้

ก.

```
x := 32.75;
y := trunc (x);
z := round (x);
WRITELN (y,' ',z);
```

ข.

```
A := -4;
B := abs (A);
D := sqrt (B);
WRITELN (B, ' ', D: 4: 1);
```

4. จงบอกเอาต์พุตจากการทำคำสั่งต่อไปนี้

ก. trunc (23 / 4);

ข. round (23 / 4);

ค. sqrt (1 + 3);
ง. sqr (4 + 5);
จ. upcase ('r');
ฉ. upcase (chr (100));

5. ถ้าหากตัวแปร num1 = 8 , num2 = 10 , และ num3 = 3 จงเขียนโปรแกรมให้แสดงค่าต่าง ๆ เป็นดังตารางต่อไปนี้

Number	It s Square	Its Cube
8	64	512
10	100	1000
3	9	27

6. ถ้าหาก name1 = Mark , name2 = Herman , name3 = Claudia ;wt1 = 153 , wt2 = 87 , wt3 = 136 ; age1 = 21 ,age2 = 9 , age3 = 20 จงเขียนโปรแกรมให้แสดงผลเป็นรูปตารางดังต่อไปนี้

Name	Weight	Age
Mark	153	21
Herman	87	9
Claudia	136	20

7. จงหาค่าที่เกิดจากการกระทำนิพจน์ต่อไปนี้

ก. 5 + Sqr(3) – 4 + trunc(3.6 – 2.1)
ข. 3 * Sqrt(8 MOD 6 * 10 DIV 5)
ค. 6 + 9 * 8 DIV 2 * Round(1.362) – 2 * 3
ง. trunc(12 / 5 * Sqrt(4 + 4 * 3) / 4)