

บทที่ 10

แถวลำดับสองมิติ : แถวลำดับของแถวลำดับ (Two - Dimensional Arrays : Arrays of Arrays)

- 10.1 แถวลำดับสองมิติ : แถวลำดับของแถวลำดับ
- 10.2 การประมวลผลแถวลำดับสองมิติ
กรณีศึกษา : การวิเคราะห์ข้อมูลการขาย
- 10.3 แถวลำดับสามมิติ
- 10.4 การแก้จุดบกพร่องของแถวลำดับหลายมิติ
- 10.5 ข้อผิดพลาดร่วมของการเขียนโปรแกรม

โครงสร้างข้อมูลแถวลำดับทำให้โปรแกรมเมอร์สามารถรวบรวมสารสนเทศในการจัดการซึ่งมีความซับซ้อนมากกว่าแถวลำดับเชิงเส้น หรือแถวลำดับหนึ่งมิติที่เราได้เห็นมาแล้ว เราสามารถประกาศและใช้แถวลำดับที่มีหลายมิติ แต่ละมิติที่มีชนิดพรรณนี้แตกต่างกัน ถึงแม้ว่าแถวลำดับหลายมิติจะทำให้มีความคล่องตัวในการจัดการข้อมูลมากกว่าแถวลำดับหนึ่งมิติ ๗ ๗ สำหรับชนิดของสมาชิกยังเหมือนเดิม สมาชิกของแถวลำดับอาจเป็นชนิดอย่างง่าย หรือชนิดโครงสร้าง แต่ทุกตัวต้องเป็นชนิดเดียวกัน

แถวลำดับหลายมิติ หมายถึง แถวลำดับที่มีมิติเป็นสองหรือมากกว่าสอง (Multidimensional array is an array with two or more dimensions.)

10.1 แถวลำดับสองมิติ : แถวลำดับของแถวลำดับ (Two - Dimensional Arrays : Array of Arrays)

แถวลำดับสองมิติ หมายถึง แถวลำดับหลายมิติร่วมส่วนใหญ่ซึ่งปกติใช้เก็บสารสนเทศแทนในรูปแบบตาราง ตัวอย่างเช่น การวางแผนที่นั่งในห้องเรียน ซึ่งแสดงรายการชื่อของนักศึกษาในตำแหน่ง (แถวและที่นั่ง) ของโต๊ะของนักศึกษา (ดูรูป 10.1)

การประกาศแถวลำดับสองมิติ (Declaring Two - Dimensional Arrays)

ตัวอย่าง 10.1 ถึง 10.3 แสดงให้เห็นว่าจะประกาศแถวลำดับสองมิติอย่างไร

ตัวอย่าง 10.1

type

MoveRange = 1..3;

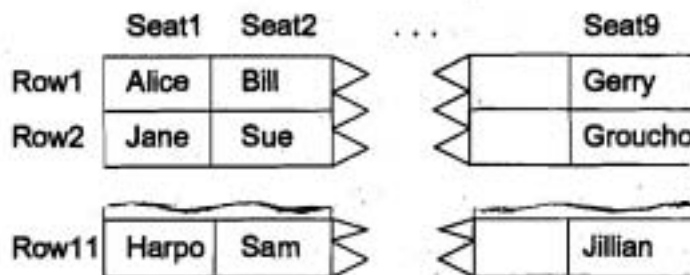
BoardArray = array [MoveRange, MoveRange] of Char;

↑ ↑
 First dimension Second dimension

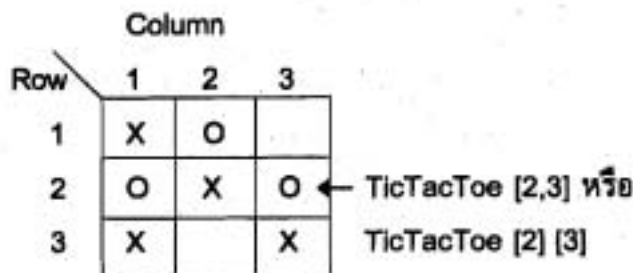
var

TicTacToe : BoardArray;

จัดสรรหน่วยเก็บสำหรับแถวลำดับ TicTacToe (ดูรูป 10.2) ซึ่งเป็นชนิด Board Array แถวลำดับชนิด BoardArray มีสองมิติ แต่ละมิติมีชนิดคระชนีล่าง เป็น MoveRange เพราะฉะนั้น TicTacToe เป็นแถวลำดับชนิดสองมิติ มี 3 แถวและ 3 สดมภ์ ดังนั้นแถวลำดับชุดนี้จึงมีสมาชิก 9 ตัว แต่ละตัวถูกอ้างถึงโดยการกำหนดคระชนีล่างของแถว (1, 2 หรือ 3) และคระชนีล่างของสดมภ์ (1, 2 หรือ 3)

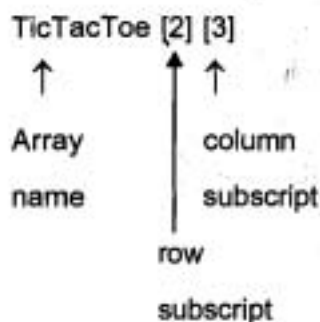
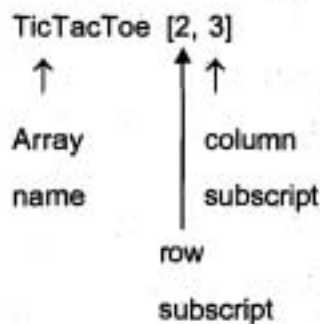


รูป 10.1 การวางแผนจัดที่นั่งในห้องเรียน (มี 11 แถว และในหนึ่งแถวมี 9 ที่นั่ง)



รูป 10.2 กระดาน TicTacToe

การอ้างถึงสมาชิกของแถวลำดับสองมิติ เราต้องระบุชื่อแถวลำดับ และจัดนิพจน์
 ครอบงำสองตัว ครอบงำลำดับตัวแรกคือครอบงำแถว (row subscript) และครอบงำลำดับที่สองคือ
 ครอบงำสดมภ์ (column subscript) ตัวแปรครอบงำสองตัวข้างล่างนี้



เลือกสมาชิกแถวลำดับในแถวที่ 2, สดมภ์ที่ 3 ของแถวลำดับ ในรูป 10.2 ซึ่ง
 ประกอบด้วยตัวอักษร 0 รูปแบบ TicTacToe[2,3] ใช้รวมมากกว่า

ตัวอย่าง 10.2

ต้องการเก็บแผนการจัดที่นั่งห้องเรียน (รูป 10.1) ในคอมพิวเตอร์ การประกาศ

const

RowMax = 11; {number of rows in the room}
 seatsPerRow = 9; {number of seats in a row}
 StringSize = 10; {characters in a string}

type

RowRange = 1 .. RowMax;
 SeatRange = 1 .. SeatsPerRow;
 StringType = string [StringSize];
 SeatPlan = array [RowRange, SeatRange] of StringType;

var

```
MyClass : SeatPlan; {seating plan for MyClass}
```

จัดสรรหน่วยเก็บสำหรับแถวลำดับสองมิติของสายอักขระ ชื่อ MyClass แถวลำดับ MyClass จะใช้เก็บชื่อแรกของที่นั่งนักศึกษาในห้องเรียนที่มี 11 แถวและในแต่ละหนึ่งแถวมี 9 ที่นั่ง เนื่องจากสายอักขระแต่ละชุดประกอบด้วยตัวอักขระไม่เกิน 10 ตัว แถวลำดับ MyClass จึงต้องใช้หน่วยเก็บจำนวน 990 ไบต์ (11 x 9 x 10)

ข้อความสั่ง

```
MyClass [6, 5] := 'Marilyn';
```

ใส่ชื่อนักศึกษา Marilyn ในแถวที่ 6 ที่นั่ง 5 ตรงกลางของห้องเรียน

ตัวอย่าง 10.3

การประกาศ

const

```
NumberSalesPeople = 5; {number of salespeople}
```

type

```
People = 1 .. NumberSalesPeople;
```

```
Quarter = (Fall, Winter, Spring, Summer);
```

```
SalesArray = array [People, Quarter] of Real;
```

var

```
Sales : SalesArray;
```

จัดสรรเนื้อที่หน่วยเก็บสำหรับแถวลำดับสองมิติของเลขจำนวนจริงชื่อ Sales มิติแรกเป็นชนิด People และมิติที่สองเป็นชนิด Quarter แถวลำดับ Sales ประกอบด้วยสมาชิก 20 ตัว (5x4) ดูรูป 10.3

ในที่นี้ Sales [1, Winter] เก็บปริมาณขาย 1250.00 สำหรับพนักงานขายคนที่ 1 ปริมาณขายระหว่าง Winter ระยะเวลาสามเดือน

แถวลำดับของแถวลำดับ (Arrays of Arrays)

แถวลำดับสามารถมีสมาชิกซึ่งเป็นชนิดโครงสร้างได้ ไม่ใช่มีแค่ชนิดอย่างง่าย ตัวอย่างของแถวลำดับที่มีสมาชิกเป็นชนิดโครงสร้าง ได้แก่ แถวลำดับของลำดับ

การประกาศ

type

```
MoveRange = 1..3;
```

```
BoardRow = array [MoveRange] of Char;
```

```
BoardArray = array [MoveRange] of BoardRow;
```

var

```
TicTacToe : BoardArray;
```

จัดสรรหน่วยเก็บสำหรับแถวลำดับ TicTacToe ที่แสดงในรูป 10.2 ข้อความสั่งเหล่านี้ประกาศ BoardRow เป็นชนิดแถวลำดับที่มีสมาชิกสามตัวชนิด Char และ BoardArray เป็นชนิดแถวลำดับ มีสมาชิกสามตัวชนิด BoardRow ดังนั้นตัวแปร TicTacToe (ชนิด BoardArray) คือแถวลำดับของแถวลำดับ หรือแถวลำดับสองมิติ

Syntax Display

การประกาศชนิดแถวลำดับ (หลายมิติ) (Array Type Declaration (Multidimensional))

Form : type

```
multidim = array [subscript1, subscript2, . . . , subscriptn] of  
element type;
```

	[Fall]	[Winter]	[Spring]	[Summer]
[1]	1200.00	1250.00	1700.00	1200.00
[2]	3500.00	2400.00	1500.00	1000.00
[3]	4500.00	3000.00	2500.00	1500.00
[4]	2500.00	2000.00	3000.00	1200.00
[5]	1250.00	2500.00	2700.00	1500.00

รูป 10.3 Array Sales

หรือ

type

multidim = array [subscript₁] of array [subscript₂]

...

of array [subscript_n] of element type;

ตัวอย่าง

type

Month = (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct,
Noy, Dec);

Candidate = 'A' .. 'E';

Precinct = 1 .. 10;

YearRange = 1900 .. 1999;

YearByMonth = array [YearRange, Month] of Real;

Election = array [candidate] of array [Precinct] of Integer;

มีความหมายดังนี้ : subscript, แทนชนิดตรรกษีล่างของมิติ i ของแถวลำดับชนิด multidim ชนิดตรรกษีล่างแต่ละตัวอาจจะเป็นชนิดเชิงอันดับที่ (ordinal) มาตรฐาน Boolean หรือ Char ชนิด enumerated หรือชนิด subrange

element type อาจจะเป็นชนิดข้อมูลมาตรฐานหรือชนิดอย่างง่ายซึ่งให้นิยามก่อนหน้าแล้ว หรือแบบชนิดข้อมูลเชิงโครงสร้าง

หน่วยเก็บของแถวลำดับสองมิติ (Storage of Two-Dimensional Arrays)

คอมไพเลอร์ของ Pascal เก็บแถวลำดับสองมิติในเซลล์หน่วยความจำประชิดกัน (adjacent memory cell) เพื่อให้เข้าถึงสมาชิกแต่ละตัวทำได้ง่าย สมาชิกของแถวลำดับสองมิติปกติเก็บเรียงอันดับตามแถว (in order by row) เช่น ชั้นแรกเก็บแถวที่ 1 จากนั้นเก็บแถวที่ 2 เช่นนี้เรื่อยไป

กระบวนการนี้เรียกว่า เรียงอันดับตามแถว (row-major order)

การเข้าถึงสมาชิกแถวลำดับตัวเฉพาะ คอมไพเลอร์จะคำนวณออฟเซต (offset) ของสมาชิกตัวนั้นจากการเก็บสมาชิกตัวแรก

การคำนวณนี้คอมพิวเตอร์ต้องทราบขนาดของไบต์ของสมาชิกแต่ละตัวและจำนวนสมาชิกในหนึ่งแถว ซึ่งทั้งสองค่านี้มีอยู่แล้วในการประกาศชนิดแถวลำดับ

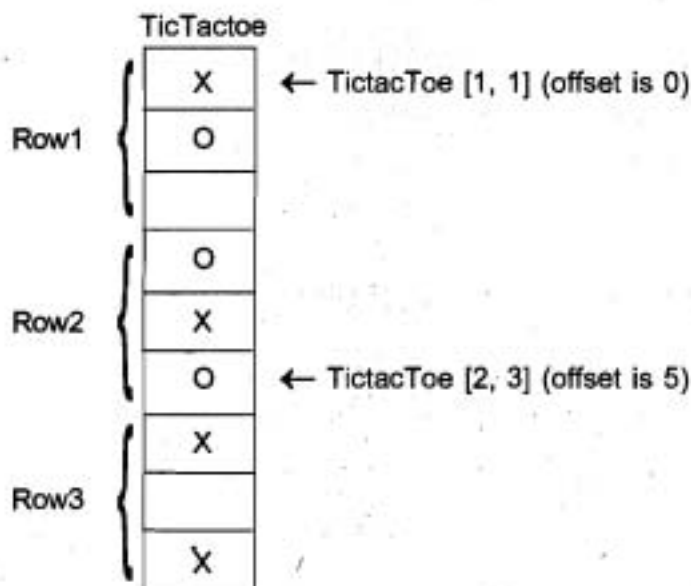
ตัวอย่างเช่น แถวลำดับ TicTacToe ซึ่งเก็บในรูป 10.4 ในแถวลำดับชุดนี้ แต่ละแถวมีสมาชิกสามตัว แต่ละตัวใช้เนื้อที่หน่วยเก็บหนึ่งไบต์ ดังนั้น ออฟเซตของสมาชิก TicTacToe [row, col]

คำนวณจากสูตร

$$\text{offset} = \text{rowsize} \times (\text{row}-1) + (\text{col}-1)$$

เพราะว่า rowsize มีค่าเท่ากับ 3 (นั่นคือ แถวหนึ่งมี 3 สดมภ์)

สูตรนี้ ให้ค่า 0 เป็น offset สำหรับสมาชิก TicTacToe [1, 1] และค่า 5 จะเป็น offset สำหรับ TicTacToe [2, 3]



รูป 10.4 แถวลำดับ TicTacToe ในหน่วยความจำ

แบบฝึกหัด 10.1 Self-Check

1. สมมติการประกาศเป็นดังนี้

type

RowRange = 1 .. 5;

ColRange = 1 .. 4;

MatrixType = array [RowRange, ColRange] of Real;

var

Matrix : MatrixType;

จงตอบคำถามต่อไปนี้

- แถวลำดับ Matrix มีสมาชิกกี่ตัว
- จงเขียนข้อความสั่งแสดงผลสมาชิกในแถวที่ 3, สดมภ์ที่ 4
- สมาชิกในข้อ b) มีค่าออฟเซต (offset) เท่าใด
- จงบอกสูตรในการคำนวณออฟเซตของ Matrix [I, J]

10.2 การประมวลผลแถวลำดับสองมิติ (Processing Two-Dimensional Arrays)

เรามีสองวิธีสำหรับการเข้าถึงสมาชิกของแถวลำดับหนึ่งมิติ : ได้แก่ เข้าถึงโดยสุ่ม และเข้าถึงแบบลำดับ สำหรับแถวลำดับสองมิติหรือตาราง เรามีสามวิธีของการเข้าถึง ได้แก่

1. เข้าถึงโดยสุ่ม (Random access) จะเรียงอันดับอย่างไรก็ได้ ขึ้นอยู่กับการใส่ข้อมูลโดยผู้ใช้โปรแกรม

2. เข้าถึงทีละแถว (Row-by row access) สมาชิกทุกตัวในแถวที่หนึ่งถูกประมวลผล จากนั้นสมาชิกทุกตัวในแถวที่สองถูกประมวลผล เช่นนี้เรื่อยไป

3. เข้าถึงทีละสดมภ์ (Column-by column access) สมาชิกทุกตัวในสดมภ์ที่หนึ่งถูกประมวลผล จากนั้นสมาชิกทุกตัวในสดมภ์ที่สองถูกประมวลผล เช่นนี้เรื่อยไป

ลูปซ้อนใน (Nested loops) ถูกนำมาใช้เพื่อเข้าถึงแถวลำดับสองมิติ อาจจะเป็นแบบทีละแถว หรือแบบทีละสดมภ์

ในการเข้าถึงทีละแถว ตัวแปรควบคุมลูปภายนอก (outer loop-control variable) คือบรรทัดนี้ล่างของแถว และตัวแปรควบคุมลูปภายใน (inner loop-control variable) คือ บรรทัดนี้ล่างของสดมภ์ บทบาทเหล่านี้จะย้อนกลับ (reversed) ในการเข้าถึงทีละสดมภ์ เช่น ที่แสดงในแผ่นแบบข้อสรุปรูปแบบของลูปข้างล่างนี้

แผ่นแบบสำหรับการเข้าถึงทีละแถวของตารางแถวลำดับ (Template for Row-by-Row Access of Array Table)

```
for row-script ... do      {Specify the row}
  for column-script ... do {Access each column of row}
    Process Table [row-script, column-script]
```

แผ่นแบบสำหรับการเข้าถึงทีละสดมภ์ของตารางแถวลำดับ (Template for Column-by Column Access of Array Table)

```
for column-script ... do  {Specify the column}
  for row-script ... do   {Access each row of column}
    Process Table [row-script, column-script]
```

ในที่นี้ คือ การดำเนินการร่วมบางชุดซึ่งเรากระทำกับแถวลำดับสองมิติ และวิธีของการเข้าถึงซึ่งแนะนำให้ใช้

- เริ่มต้นให้สมาชิกทุกตัวของแถวลำดับมีค่าซึ่งกำหนดให้ อาจจะเข้าถึงทีละแถว หรือเข้าถึงทีละสดมภ์

- หาผลบวกของสมาชิกทุกตัวในตาราง - อาจจะเข้าถึงทีละแถว หรือเข้าถึงทีละสดมภ์

- แสดงผลสมาชิกในตาราง - เข้าถึงทีละแถว

- หาผลบวกของแต่ละแถว - เข้าถึงทีละแถว

- หาผลบวกของแต่ละสดมภ์ - เข้าถึงทีละสดมภ์

- ปรับค่า (update) สมาชิกของแถวลำดับตามที่กำหนดโดยข้อมูลอินพุต - เข้าถึงโดยสุ่ม

เราจะแสดงให้เห็นว่าจะกระทำการดำเนินการเหล่านี้บางชุดได้อย่างไรในหัวข้อถัดไป การเริ่มต้นของแถวลำดับ (Initializing an Array)

กระบวนการ Initialize (รูป 10.5) เริ่มต้นสมาชิกแต่ละตัวของพารามิเตอร์แถวลำดับของมัน Sales (รูป 10.3) ด้วย InValue

กระบวนการงานเข้าถึงสมาชิกแถวลำดับที่ละแถว for ลูปชุดนอก รอบ (cycle) ของมันผ่านแถว และ for ลูปชุดใน รอบของมันผ่านสตมภ์

ระหว่างการวนซ้ำครั้งแรก (first iteration) ของลูปนอก NextPerson มีค่าเท่ากับ 1 ดังนั้น ลูปในตั้งค่าสมาชิกของแถวลำดับที่ตัวในแถวที่หนึ่ง ให้เป็น InValue ต่อไปตั้งค่าสมาชิกของแถวลำดับที่ตัวในแถวที่สอง ให้เป็น InValue เช่นนี้เรื่อยไป

```
procedure Initialize (var Sales {output} : SalesArray; InValue {input} : Real);
{
  Initializes all elements of its array parameter to InValue.
  Pre : InValue is defined.
  Post : Sets each element of its array parameter to InValue.
}
var
  NextPerson : People; {Row subscript}
  NextQuarter : Quarter; {Column subscript}

begin {Initialize}
  for NextPerson := 1 to NumberSalesPeople do
    Sales [NextPerson, NextQuarter] := InValue
end; {Initialize}
```

รูป 10.5 การเริ่มต้นของแถวลำดับสองมิติ

การคำนวณผลบวกของแถว (Computing Row Sums)

สำหรับแถวลำดับซึ่งแสดงในรูป 10.3 การคำนวณผลบวกของหนึ่งแถวจะให้ปริมาณขายทั้งหมดรายปีสำหรับพนักงานขายหนึ่งคน

(ตัวอย่างเช่น $1200.00 + 1250.00 + 1700.00 + 1200.00$ คือปริมาณขายทั้งหมดในหนึ่งปีของพนักงานขายคนที่หนึ่ง)

ส่วนของโปรแกรม

```
.Sum := 0.0;  
for NextQuarter := Fall to Summer do  
    Sum := Sum + Sales [1, NextQuarter];  
WriteLn ('Sales for person 1 are $ ', Sum : 3 : 2)
```

คำนวณผลบวกนี้โดยการบวกสมาชิกแถวลำดับสี่ตัวในแถวที่ 1 กับ Sum โปรดสังเกตว่าเฉพาะตรรกะนี้ล่างของสคีมเท่านั้น ที่เปลี่ยนแปลงในการวนซ้ำลูปแต่ละรอบ การเริ่มต้นให้ Sum มีค่าเท่ากับ 0.0 กระทำก่อนเข้าลูป และหลังจากออกจากลูป แสดงผลค่าสุดท้ายของ Sum

เราต้องใช้ลูปซ้อนในเพื่อหาผลบวกของแต่ละแถว กระบวนการ SumRows (รูป 10.6) กระทำการดำเนินการนี้แทนที่จะแสดงผลบวกแต่ละแถว กระบวนการกลับคืนผลบวกสำหรับแถวที่ I ในสมาชิกแถวลำดับ RowSum (I) ชนิด RowArray เราประกาศชนิด Row Array ก่อนกระบวนการ

กระบวนการ SumRows เข้าถึงสมาชิกแถวลำดับที่ละแถว for ลูปชุดนอกวนรอบผ่านแถว และ for ลูปชุดในของมีนวนรอบผ่านสคีม ลูปนอกเริ่มต้นให้ Sum มีค่าเท่ากับ 0.0 ก่อนผ่านแต่ละครั้งตลอดสคีม และกำหนดให้ค่าสุดท้ายของ Sum หลังจากผ่านแต่ละครั้งให้กับสมาชิกของแถวลำดับ SumRow ด้วยตรรกะนี้ล่างของแถว

type

```
RowArray = array [People] of Real;
```

```
procedure SumRows (Sales {input} : SalesArray; var RowSum {output} :  
    RowArray);
```

```
{
```

```
    Sums the row of array Sales, returning the sums in RowSum.
```

```
    Pre : Array Sales is defined.
```

```
    Post : RowSum [I] is the sum of elements in row I.
```

```
}
```

```

var
    NextPerson : People; {Rows cript}
    NextQuarter : Quarter; {Column script}
    Sum : Real;          {Each row sum}

begin {SumRows}
    for NextPerson := 1 to NumberSalesPeople do
        begin
            Sum := 0.0;
            for NextQuarter := Fall to Summer do
                Sum := Sum + Sales [NextPerson, NextQuarter];
            RowSum [NextPerson] := Sum {Return next row sum}
        end; {outer for}
    end, {SumRows}

```

รูป 10.6 กระบวนการ SumRows และชนิด RowArray

การคำนวณผลบวกของสดมภ์ (Computing Column Sums)

การคำนวณผลบวกของแต่ละสดมภ์จะให้ปริมาณขายทั้งปีของแต่ละไตรมาส (quarter) ตัวอย่างเช่น 1200.00 + 3500.00 + 4500.00 + 2500.00 + 1250.00 ให้ผลรวมยอดขายทั้งหมดของไตรมาส Fall การคำนวณผลบวกสำหรับสดมภ์ที่กำหนดให้ เราต้องบวกสมาชิกห้าตัวในแต่ละแถวของสดมภ์นั้น

กระบวนการ SumColumns (รูป 10.7) เข้าถึงสมาชิกแถวลำดับที่แต่ละสดมภ์ และกลับคืนผลบวกสดมภ์ที่ตัวผ่านแถวลำดับ ColSmm รูป 10.8 แสดงความสัมพันธ์ระหว่างแถวลำดับ Sales, RowSum และ ColSum

```

type
    ColArray = array [Quarter] of Real;

procedure SumColumns (Sales {input} : SalesArray;
                    var ColSum {output} : ColArray);
{
    Sums the columns of array Sales, returning the sums in ColSum
    Pre : Array Sales is defined.
    Post : ColSum [I] is the sum of elements in column I.
{
    var
        NextPerson : People; {Row subscript}
        NextQuarter : Quarter; {Column subscript}
        Sum : Real;          {Each column sum}

begin {SumColumns}
    for NextQuarter := Fall to Summer do
        begin
            Sum := 0.0
            for NextPerson := 1 to NumberSalesPeople do
                Sum := Sum + Sales [NextPerson, NextQuarter];
            ColSum [NextQuarter] := Sum {Return next column sum}
        end {outer for}
    end, {SumColumns}

```

รูป 10.7 กระบวนงาน SumColumns และชนิด ColArray

	[Fall]	[Winter]	[Spring]	[Summer]	Person Totals
[1]	1200.00	1250.00	1700.00	1200.00	→RowSum [1]
[2]	3500.00	2400.00	1500.00	1000.00	→RowSum [2]
[3]	4500.00	3000.00	2500.00	1500.00	→RowSum [3]
[4]	2500.00	2000.00	3000.00	1200.00	→RowSum [4]
[5]	1250.00	2500.00	2700.00	1500.00	→RowSum [5]
Quarterly	ColSum	ColSum	ColSum	Colsum	
Totals	[Fall]	[Winter]	[Spring]	[Summer]	

รูป 10.8 แถวลำดับ Sales, RowSum และ ColSum

กรณีศึกษา การวิเคราะห์ข้อมูลขาย (Analysis of Sales Data)

1. ปัญหา (Problem)

ผู้จัดการแผนกขายของบริษัทต้องการโปรแกรมวิเคราะห์การขายเพื่อติดตาม (track) ความสามารถขายของพนักงานขายจำแนกตามไตรมาส (quarter) โปรแกรมจะอ่านธุรกรรมการขายทั้งหมดจากแฟ้มข้อความ ข้อมูลของธุรกรรมแต่ละชุดจะมีเลขประจำตัวของพนักงานขาย ไตรมาสซึ่งเกิดการขาย และปริมาณขาย

ธุรกรรมการขาย (sales transaction) ไม่จำเป็นต้องเรียงอันดับหลังจากอ่านธุรกรรมขายทั้งหมดแล้ว โปรแกรมจะแสดงตารางในรูปแบบที่แสดงในรูป 10.8 ซึ่งรวมทั้งผลรวมทั้งหมดของพนักงานขายและจำแนกตามไตรมาส

2. วิเคราะห์ (Analysis)

เราจำเป็นต้องแยกแถวลำดับเพื่อเก็บตารางข้อมูลการขาย ผลบวกแถวและผลบวกสตมภ์

ข้อมูลที่ต้องการ (Data Requirements)

อินพุตปัญหา (Problem Inputs)

แฟ้มรายการเปลี่ยนแปลงการขาย (Sales transaction file)

Sales : SalesArray ; {array of sales amounts}

เอาต์พุตปัญหา (Problem Outputs)

RowSum : RowArray ; {array of row sums}

ColSum : ColArray ; {array of column sums}

3. ออกแบบ (Design)

Initial Algorithm

1. เริ่มต้นให้สมาชิกทุกตัวของแถวลำดับ Sales มีค่าเป็นศูนย์
2. อ่านข้อมูลขาย
3. คำนวณผลบวกแถว
4. คำนวณผลบวกสดมภ์
5. แสดงผลตารางการขายที่มีผลบวกแถวและผลบวกสดมภ์

4. การทำให้เกิดผล (Implementation)

ลงรหัสโปรแกรมหลัก (Coding the Main Program)

เราได้เขียนกระบวนการงานสำหรับขั้นที่ 1, 3 และ 4 เสร็จแล้ว (รูป 10.5 - รูป 10.7)

ต่อไปเราจะเขียนโปรแกรมหลัก (รูป 10.9) หลังจากนั้นเขียนกระบวนการงานสำหรับขั้นที่ 2 และขั้นที่ 5

{SR+}

program SalesAnalysis;

}

Read in sales figures for one year and stores them in a table organized by salesperson and quarter.

Displays the table and the annual totals for each person and the sales totals for each quarter.

}

const

NumberSalesPeople = 5; {number of salespeople}

type

```

People = 1 .. NumberSalesPeople;
Quarter = (Fall, Winter, Spring, Summer);
SalesArray = array [People, Quarter] of Real;
RowArray = array [People] of Real;
ColArray = array [Quarter] of Real;

```

{Insert procedure Initialize, SumRows, and SumColumns.}

{Insert procedure ReadQuarter, ReadTable and DisplayTable.}

```

begin {SalesAnalysis}
  Initialize (Sales, 0.0);
  ReadTable (Sales);
  SumRows (Sales, RowSums);
  SumColumns (Sales, ColSums);
end. {SalesAnalysis}

```

รูป 10.9 โปรแกรมหลักของการวิเคราะห์การขาย

การลงรหัสกระบวนการ ReadTable (Coding Procedure ReadTable)

กระบวนการ ReadTable (รูป 10.10) อ่านข้อมูลตารางการขายจากแฟ้มรายการเปลี่ยนแปลงการขาย หลังจากมีการเตรียมเพิ่มข้อมูลให้เป็นอินพุต รายการเปลี่ยนแปลงแต่ละชุดกำหนดพนักงานขายไตรมาสซึ่งเกิดการขาย และปริมาณขาย กระบวนการ ReadQuarter (ดูแบบฝึกหัดข้อ 1 ที่ตอนจบของหัวข้อนี้) กลับคืนค่ายอดขายรายไตรมาสซึ่งอ่านตลอดพารามิเตอร์ตัวที่สองของมัน และค่าแบบบูลีน (Boolean value) ตลอดพารามิเตอร์ตัวที่สามของมัน ซึ่งแสดงว่าไตรมาสซึ่งอ่านนั้นถูกต้องหรือไม่

ข้อความสั่ง if ทดสอบว่า พนักงานขาย และไตรมาสถูกต้องหรือไม่ ถ้าถูกต้องข้อความสั่ง

Sales [TranPerson, TranQuarter] :=

Sales [TranPerson, TranQuarter] + TranAmount

บวกปริมาณขายปัจจุบันกับผลรวมให้สะสมสำหรับพนักงานขายซึ่งเลือกโดย TranPerson (คหกรรชนี่ล่งแถว) และ TranQuarter (คหกรรชนี่ล่งสคคมกร์) เนื่องจกข้อมูลไม่ เรียงอ้นด้น สมหกรรชแถวล่งด้นจ้งถูกเข้ถึงโดยสุม

```
procedure ReadTable (var Sales {output} : SalesArray);
```

```
{
```

Reads the sales table data from a data file and saves the sales results in a sale table.

Pre : Sales is initialized to all zeros.

Post : Each entry of Sales represents the sum of sales for a particular salesperson and quarter.

Calls : ReadQuarter to read the sales quarter.

```
}
```

```
var
```

```
TranAmount : Real; {transaction amount}
```

```
TranPerson : Integer; {salesperson number}
```

```
TranQuarter : Quarter; {sales quarter}
```

```
SalesFile : Text; {sales file variable}
```

```
FileName : String; {external file name}
```

```
ValidQuarter : Boolean; {flag indicating whether a valid quarter was read}
```

```
begin {ReadTable}
```

```
{Prepare data file for input}
```

```
Write ('Enter name of sales data file> ');
```

```
ReadLn (FileName);
```

```

Assign (SalesFile, FileName);
Reset (SalesFile);

{Read and store all valid sales data}
while not EOF (SalesFile) do
  begin
    Read (SalesFile, TranPerson);
    ReadQuarter (SalesFile, TranQuarter, ValidQuarter);
    ReadLn (SalesFile, TranAmount);
    if ValidQuarter and (TranPerson >= 1) and (TranPerson <= 5) then
      Sales [TranPerson, TranQuarter] :=
        Sales [TranPerson, TranQuarter] + TranAmount
    else
      WriteLn ('Invalid person or Quarter-', 'person is', TranPerson : 1,
        '; amount is ', TranAmount : 3 : 2)
    end {while}
end; {ReadTable}

```

รูป 10.10 กระบวนการ ReadTable

การลรหัสกระบวนการ DisplayTable (Coding Procedure DisplayTable)

เราควรแสดงผลสารสนเทศ ในแถวลำดับสองมิติวิธีเดียวกับที่มนุษย์มองเห็นมัน คือเป็นตารางซึ่งแถวสมนัยกับมิติแรกของแถวลำดับ และสดมภ์ของมันสมนัยกับมิติที่สองของแถวลำดับ การทำสิ่งนี้ให้ประสบผลสำเร็จ คือ เข้าถึง และแสดงผลสมาชิกของแถวลำดับทีละแถว (row-by-row)

กระบวนการ DisplayTable (รูป10.11) แสดงข้อมูลในแถวลำดับตารางขาย ในรูปแบบที่แสดงให้เห็น ในรูป 10.8 นอกจากข้อมูลแถวลำดับ (พารามิเตอร์ตัวแรก) แล้ว กระบวนการยังแสดงผลผลบวกแถว (พารามิเตอร์ตัวที่สอง) และผลบวกสดมภ์ (พารามิเตอร์

ตัวที่สาม) เพื่อสนับสนุนการอ่านง่าย กระบวนงาน DisplayTable ยังแสดงผลตารางแถว และตารางสทมภ์ แสดงให้เห็นในรูป 10.8

```
.procedure Display Table (Sales {input} : SalesArray;
                          RowSum {input} : RowArray;
                          ColSum {input} : ColArray);
{
  Displays the sales table data in table form and the row and column sums.
  Pre : Sales, RowSum, and ColSum are defined.
  Post : The values stored in the three arrays are displayed.
}
var
  NextPerson : People; (Each salesperson)
  NextQuarter : Quarter; (Each quarter)

begin
  {Display the heading.}
  WriteLn;
  WriteLn ('SALES TABLE' : 40);
  WriteLn ('[Fall]' : 20, '[Winter]' : 10, '[Spring]' : 10, '[Summer]' : 10,
          'Person Totals');

  {Display one table row for each salesperson.}
  for NextPerson := 1 to NumberSalesPeople do
    begin
      Write ('[' : 8, NextPerson : 1, '[');
      for NextQuarter := Fall to Summer do
        Write (Sales [NextPerson, NextQuarter] : 10 : 2);
```

```

        WriteLn (RowSum [NextPerson] : 14 : 2)
    end; {for NextPerson}

{Display the column totals.}
WriteLn;
Write ('Quarterly' : 10);
for NextQuarter := Fall to Summer do
    Write (ColSum [NextQuarter] : 10 : 2);
WriteLn;
WriteLn ('Totals')
end; {DisplayTable}

```

รูป 10.11 กระบวนการ DisplayTable

5. ทดสอบ (Testing)

การทดสอบโปรแกรมวิเคราะห์การขาย ให้สร้างเพิ่มข้อมูลตัวอย่างตรวจทานว่า ข้อมูลไม่ถูกต้องสำหรับพนักงานขายหรือไตรมาสไม่ทำให้เกิด Range check error ระหว่างเวลาวิ่งโปรแกรม รูป 10.12 แสดงให้เห็นการวิ่งตัวอย่าง (sample run) สดมภ์รวมทั้งหมด รายพนักงานขายและแถวรวมทั้งหมดรายไตรมาส ทั้งคู่นี้ต้องเป็นจำนวนเดียวกัน (4200.00 สำหรับการวิ่งตัวอย่าง)

```

Enter name of sales data fiel > SALEDATA.TXT
Invalid person or quarter - person is 6 ; amount is 1200.00
Invalid person of quarter - person is 3 ; amount is 850.00

```

SALES TABLE

	[Fall]	[Winter]	[Spring]	[Summer]	Person Totals
[1]	1000.00	0.00	0.00	1500.00	2500.00
[2]	00.00	0.00	100.00	500.00	600.00
[3]	100.00	0.00	1000.00	00.00	1100.00
[4]	0.00	0.00	0.00	0.00	0.00
[5]	0.00	0.00	0.00	0.00	0.00
Quarterly	1100.00	0.00	1100.00	2000.00	
Totals					

รูป 10.12 การวิ่งตัวอย่างของโปรแกรมวิเคราะห์การขาย

แบบฝึกหัด 10.2 Self-Check

1. สำหรับส่วนของโปรแกรมแต่ละชุด ตำแหน่งแถวลำดับ (array locations) ชุดใดบ้างถูกแสดงผลและเรียงอันดับอย่างไร

- a) for NextQuarter := Fall to Summer do
Write (Sales [1, NextQuarter])
- b) for NextPerson := 1 to 5 do
Write (Sales [NextPerson, Spring]);
- c) for NextQuarter := Fall to Summer do
begin
for NextPerson := 1 to 5 do
Write (Sales [NextPerson, NextQuarter]);
WriteLn
end; {for}

เขียนโปรแกรม

1. กระบวนงาน ReadQuarter อ่านตัวอักษรสองตัวแทนไตรมาส (FA, WI, SP, SU) จากแฟ้มข้อมูลและกลับคืนค่าไตรมาสที่สมนัยกัน (Fall, Winter, Spring, Summer) ถ้าตัวอักษรข้อมูลถูกต้อง จงเขียนกระบวนงาน ReadQuarter

2. จงเขียนการประกาศชนิดแถวลำดับสำหรับเมทริกซ์จัตุรัสขนาด 10 x 10 ซึ่งประกอบด้วยเลขจำนวนจริง

3. สำหรับการประกาศชนิดแถวลำดับในแบบฝึกหัดข้อ 2 จงทำสิ่งต่อไปนี้

a) จงเขียนลูปซึ่งคำนวณผลบวกของสมาชิกในแถวที่ 5

b) จงเขียนลูปซึ่งคำนวณผลบวกของสมาชิกในสตมภ์ที่ 6

c) จงเขียนลูปซ้อนใน (nested loops) ซึ่งคำนวณผลบวกของสมาชิกแถวลำดับ
ทุกตัว

4. จงเขียนฟังก์ชันซึ่งตรวจสอบว่าใครเป็นคนชนะเกม tic-tac-toe ฟังก์ชันนี้ขั้นแรกตรวจสอบทุกแถวเพื่อดูว่ามีผู้เล่นหนึ่งคนครอบครองทุกเซลล์ในแถวนั้นหรือไม่ ต่อไปตรวจสอบทุกสตมภ์ จากนั้นตรวจสอบเส้นทแยงมุมทั้งสองด้าน ฟังก์ชันนี้กลับคืนหนึ่งค่าชนิด enumerated (Nowinner, XWins, OWins)

10.3 แถวลำดับสามมิติ (Three-Dimensional Arrays)

Pascal ไม่มีการจำกัดจำนวนมิติที่แถวลำดับอาจจะมี แต่แถวลำดับที่ใช้ร่วมส่วนใหญ่มีเฉพาะสองมิติและสามมิติเท่านั้น

เราต้องทราบปริมาณใหญ่ของเนื้อที่หน่วยความจำที่จะถูกจัดสรรสำหรับการเก็บแถวลำดับหลายมิติ สำหรับเหตุผลนี้จึงหลีกเลี่ยงการส่ง (passing) แถวลำดับหลายมิติเป็นพารามิเตอร์ค่า ในหัวข้อนี้เราศึกษาตัวอย่างของแถวลำดับหลายมิติ

ตัวอย่าง 10.4

แถวลำดับ RoomHeat (รูป 10.13) ประกาศดังนี้

type

Length = 1 .. 12; {12 feet long}

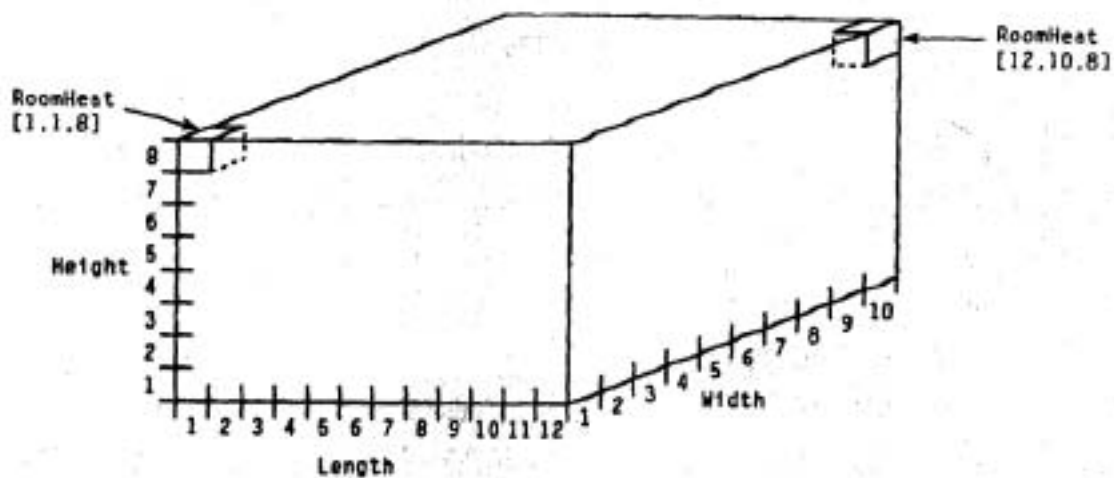
Width = 1 .. 10; {10 feet wide}

Height = 1 .. 8; {8 feet high}

TempArray = array [Length, Width, Height] of Real;

var

RoomHeat : TempArray;



รูป 10.13 Array RoomHeat

มีสามมิติและต้องใช้บรรทัดสามตัว บรรทัดสามตัวแรก มีค่าจาก 1 ถึง 12 บรรทัดสามตัวที่สองมีค่าจาก 1 ถึง 10 และบรรทัดสามตัวที่สามมีค่าจาก 1 ถึง 8 เพราะฉะนั้นจึงมีเลขจำนวนจริงที่เก็บในแถวลำดับ RoomHeat ทั้งหมด $12 \times 10 \times 8 = 960$ ตัวเราสามารถใช้อแถวลำดับนี้เพื่อเก็บอุณหภูมิห้องวัดที่จุดไม่ต่อเนื่อง (discrete points) ในห้องขนาด 12×10 ฟุต ซึ่งความสูงถึงเพดานเท่ากับ 8 ฟุต

ตัวอย่าง 10.5

งานของเราคือทำให้เป็นระบบคอมพิวเตอร์ (computerize) ในสำนักงานขนาดเล็ก ชั้นที่หนึ่งเพื่อเก็บสารสนเทศในตู้เก็บแฟ้มสองตู้ของเจ้านาย คิตเปียย "Business" และ "Personal" (รูป 10.14) ตู้เก็บแฟ้มแต่ละตู้มีสองลิ้นชัก แต่ละลิ้นชักมี 28 โฟลเดอร์ (folders) คิตเปียย A ถึง Z แต่ละโฟลเดอร์ประกอบด้วยบันทึก (memo) สูงสุด 255 ตัวอักษร การประกาศต่อไปนี้จัดสรรหน่วยเก็บสำหรับแถวลำดับ Office ซึ่งจะเก็บสารสนเทศทั้งหมดในตู้เก็บแฟ้มสองตู้

```
const
    MemoLength = 255;
```

type

CabinetType = (Business, Personal);

DrawerType = (Top, Bottom);

FolderType = 'A' ... 'Z';

Memo = string [MemoLength];

FileCabinet = array [CabinetType, DrawerType, FolderType] of Memo;

var

Office : FileCabinet;

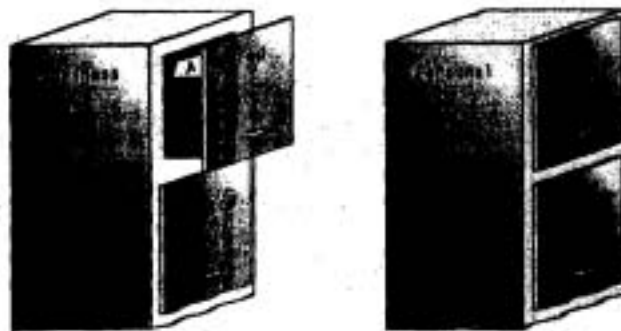
เราต้องใช้บรรทัดสามตัวเพื่อเข้าถึงสมาชิกของแถวลำดับชุดนี้ ตัวแปรบรรทัด
ล่าง Office [Business, Top, 'A'] เข้าถึงโฟลเดอร์ที่มีป้ายติดว่า A ในลิ้นชักเปิดซึ่งแสดงใน
รูป 10.14

ข้อความสั่ง

Write (MemoFile, Office [Business, top, 'A'])

ทำสำเนาบันทึก (memo) ในโฟลเดอร์นี้ไปยังแฟ้มเอาต์พุตชื่อ MemoFile (ชนิด
Text)

แถวลำดับ Office จะใช้เนื้อที่หน่วยเก็บปริมาณมาก ให้ใช้ 1K ไบต์ หมายถึง 1024
ไบต์ แถวลำดับชุดนี้ใช้หน่วยเก็บ = $2 \times 2 \times 26 \times 256$ ไบต์
= 26 K ไบต์



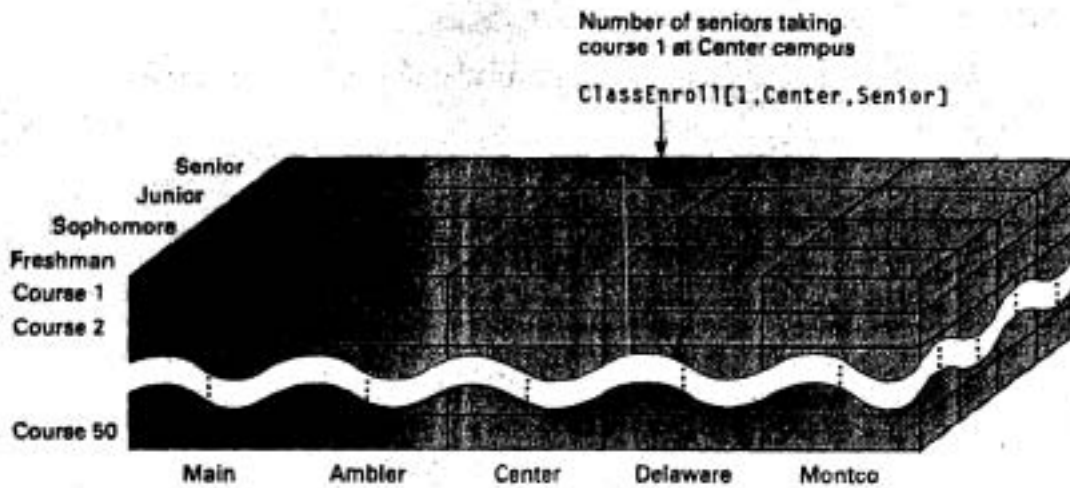
รูป 10.14

ตัวอย่าง 10.6

มหาวิทยาลัยแห่งหนึ่งมีวิทยาเขต (campuses) ห้าแห่ง วิทยาเขตแต่ละแห่งเปิดสอน 50 กระบวนวิชา (courses) สำนักงานลงทะเบียนต้องการแบ่งสารสนเทศการลงทะเบียนให้ เด็กกลง โดยขึ้นอยู่กับชั้นปีของนักศึกษา (student rank) การทำสิ่งนี้ เขาจำเป็นต้องเก็บ สารสนเทศของการลงทะเบียนในแถวลำดับสามมิติที่มีสมาชิก $50 \times 5 \times 4 = 1000$ ตัว ต่อไปประกาศแถวลำดับชุดนี้ และแสดงให้เห็นในรูป 10.15

const

MaxCourse = 50; (maximum number of courses)



รูป 10.15 Three-Dimensional Array ClassEnroll

type

Campus = (Main, Ambler, Center, Delaware, Montco);

Rank = (Freshman, Sophomore, Junior, Senior);

CourseRange = 1 . . MaxCourse;

ClassArray = array [CourseRange, Campus, Rank] of Integer;

var

ClassEnroll : ClassArray; {class enrollment}

CurCampus : Campus; {current campus}

ClassRank : Rank; {current rank}

Total : Integer; {student total}

ตัวแปรธรรมณี ClassEnroll [1, Center, Senior] แทนจำนวนนักศึกษาชั้นปีที่ 4
ลงทะเบียนวิชาที่ 1 ที่วิทยาเขต Center

ส่วนของโปรแกรม

Total := 0

for ClassRank := Freshman to Senior do

total := total + ClassEnroll [1, Center, ClassRank]

คำนวณจำนวนนักศึกษาทั้งหมดทุกชั้นปีในวิชาที่ 1 ที่วิทยาเขต Center

ส่วนของโปรแกรม

Total := 0 ;

for CurCampus := Main to Montoo do

for ClassRank := Freshman to Senior do

Total := Total + ClassEnroll [1, Curcampus, ClassRank]

คำนวณจำนวนนักศึกษาทั้งหมดในวิชาที่ 1 (ไม่สนใจชั้นปีหรือวิทยาเขต)

แบบฝึกหัด 10.3 Self-Check

1. จงประกาศชนิดแถวลำดับสามมิติ ซึ่งธรรมณีล่างตัวแรกประกอบด้วยตัวอักษร
จาก 'A' ถึง 'F' ธรรมณีล่างตัวที่สองประกอบด้วยจำนวนเต็มจาก 1 ถึง 10 และธรรมณีล่าง
ตัวที่สามประกอบด้วย ชนิด Day (วันของสัปดาห์) ซึ่งนิยามโดยผู้ใช้ เลขจำนวนจริงจะถูก
เก็บในแถวลำดับชุดนี้ จากนั้นจงคำนวณว่าแถวลำดับชนิดนี้จะเก็บสมาชิกได้ทั้งหมดกี่ตัว

2. จงประกาศแถวลำดับสามมิติซึ่งเก็บจำนวนนักศึกษาในชั้นเรียนวิชาคณิตศาสตร์
(Math1, Algebra1, Geometry, Algebra2, Trigonometry, Calculus) ที่โรงเรียนมัธยมเก่า
ของเรา จำแนกประเภทตามระดับเกรด และเพศ ของนักศึกษา จากนั้นให้คำนวณว่า แถว
ลำดับชุดนี้จะมีสมาชิกทั้งหมดกี่ตัว

3. จงขยายการเก็บทีละแถว (row-major order) ของสามมิติและแสดงให้เห็นว่า แถวลำดับ ClassEnroll ถูกเก็บอย่างไร และจงคำนวณหา ออฟเซต (offset) ของสมาชิก แถวลำดับ ClassEnroll [1, Center, Senior] และสูตรทั่วไป สำหรับ ClassEnroll [i, j, k]

เขียนโปรแกรม

1. จงให้นิยามใหม่ (redefine) MasCourse (ตัวอย่าง 10.6) ให้เท่ากับ 5 และเขียน ส่วนของรหัส (code fragments) ซึ่งกระทำการดำเนินการต่อไปนี้

a) ใส่ข้อมูลการลงทะเบียน

b) จงคำนวณหาจำนวนนักศึกษาชั้นปีที่สาม (juniors) ในทุกชั้นเรียนและทุก วิทยาเขต จากนั้นให้นำจำนวนนักศึกษาซึ่งลงทะเบียนเพียงหนึ่งวิชาเท่านั้น

c) จงนับจำนวนนักศึกษาชั้นปีที่สอง (sophomores) ในทุกวิทยาเขต ซึ่งลงทะเบียนวิชาที่ 2

d) จงคำนวณและพิมพ์จำนวนนักศึกษาที่วิทยาเขต Main ซึ่งลงทะเบียนในแต่ละวิชาและจำนวนนักศึกษาทั้งหมดที่วิทยาเขต Main ในทุกวิชา นับจำนวนนักศึกษาซึ่งลงทะเบียนเพียงหนึ่งวิชา

e) จงคำนวณและพิมพ์จำนวนนักศึกษา upper-class ในทุกวิชาที่แต่ละวิทยาเขต เช่นเดียวกับจำนวนทั้งหมดของนักศึกษา upper-class ที่ลงทะเบียน (Upper-class students หมายถึงนักศึกษาชั้นปีที่สามและชั้นปีที่สี่) จากนั้นให้นำจำนวนนักศึกษาซึ่งลงทะเบียนเพียงหนึ่งวิชาเท่านั้น

10.4 การแก้จุดบกพร่องของแถวลำดับหลายมิติ (Debugging Multidimensional Arrays)

เราสามารถใช้นำต่าง Watch เพื่อแสดงผลสมาชิกแต่ละตัวของแถวลำดับหลายมิติ สมาชิกแถวลำดับซึ่งประชิดกันหลายตัว หรือแถวลำดับหลายมิติทั้งหมด แถวลำดับหลายมิติแสดงในหน้าต่าง Watch เป็นแบบซ้อนใน รายการสมาชิกในวงเล็บ ถ้าต้องการแสดงค่าทั้งหมดในแถวลำดับสองมิติในรูป 10.2 ใช้นิพจน์ Watch ดังนี้

TicTacToe

หน้าต่าง Watch จะแสดง content ของแถวลำดับ TicTacToe เป็น TicTacToe : (('X', 'O', ' '), ('O', 'X', 'O'), ('X', ' ', 'X')) สมาชิกของ TicTacToe ปรากฏทีละแถว คือ แถวที่ 1 ตามด้วยแถวที่ 2 และแถวที่ 3 ถ้าเราใช้นิพจน์ Watch

TicTacToe [2]

สมาชิกของแถวที่สองของแถวลำดับ TicTacToe ปรากฏในหน้าต่าง Watch เป็น TicTacToe [2]: ('O', 'X', 'O')

การนับซ้ำ (Repeat counts) ในนิพจน์ Watch แสดงสมาชิกซึ่งประชิดกันหลายตัวของแถวลำดับ TicTacToe หรือแถวประชิดหลายแถวของแถวลำดับ TicTacToe นิพจน์ Watch

TicTacToe [2, 1], 2

แสดงสมาชิกตัวที่หนึ่งและตัวที่สองของแถวที่ 2 ในแถวลำดับ TicTacToe เป็น TicTacToe [2, 1], 2 : 'O', 'X'

ใช้การนับซ้ำอย่างรอบคอบ เพราะว่า สมาชิกแถวลำดับประชิดกันไม่จำเป็นต้องอยู่ในแถวเดียวกัน นิพจน์ Watch

TicTacToe [2, 3], 2

แสดงสมาชิกแถวลำดับ TicTacToe [2, 3] และ TicTacToe [3, 1] ในหน้าต่าง Watch นิพจน์ Watch

TicTacToe [1], 2

แสดงแถวที่หนึ่งและแถวที่สองของแถวลำดับ TicTacToe เป็น

TicTacToe [1], 2 : ('X', 'O', ' '), ('O', 'X', 'O')

10.5 ข้อผิดพลาดร่วมของการเขียนโปรแกรม (Common Programming Errors)

เมื่อเราใช้แถวลำดับหลายมิติ ให้มั่นใจว่าดรรชนีสร้างสำหรับแต่ละมิติต้องกันกับชนิดที่ได้ประกาศไว้ ถ้าค่าของดรรชนีสร้างใดๆ ก็ตาม อยู่นอกพิสัย เราจะได้ Range check error ระหว่างเวลาดำเนินงาน

ถ้าเราใช้ for ลูปซ้อนในเพื่อประมวลผลสมาชิกแถวลำดับ ต้องมั่นใจว่า ตัวแปรควบคุมลูป ซึ่งใช้เป็นดรรชนีสร้างแถวลำดับเรียงอันดับถูกต้อง การเรียงอันดับของตัวแปรเหล่านี้จะกำหนดลำดับซึ่งสมาชิกแถวลำดับจะถูกประมวลผล

ข้อสรุปของตัวสร้างใหม่ของ Pascal (Summary of New Pascal Constructs)

Construct	Effect
<p>การประกาศแถวลำดับหลายมิติ (Declaring Multidimensional Arrays)</p> <pre> type Day = (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday); Matrix = array [1 .. 52, Day] of Real; . var Sales : Matrix ; การอ้างถึงแถวลำดับหลายมิติ (Multidimensional Array Reference) Write (Sales [3, Monday]) for Week := 1 to 52 do for Today := Sunday to Saturday do Sales [Week, Today] := 0.0 ReadLn (Sales [1, Sunday]) </pre>	<p>Matrix อธิบายแถวลำดับสองมิติที่มี 52 แถว และ 7 สดมภ์ (วันของ สัปดาห์)</p> <p>Sales เป็นแถวลำดับชนิด Matrix และสามารถเก็บเลขจำนวนจริงได้ $52 \times 7 = 364$ ตัว</p> <p>แสดงผลสมาชิกของ Sales สำหรับ วัน Monday ของสัปดาห์ที่ 3</p> <p>กำหนดค่าเริ่มต้นของสมาชิกทุกตัว ของ Sales ให้เท่ากับ 0.0</p> <p>อ่านค่าสำหรับ Sunday แรกไว้ใน Sales</p>

แบบฝึกหัด Quick-Check

1. ภาวะการเข้าถึง (modes of access) แถวลำดับสองมิติมี 3 ชนิดอะไรบ้าง
2. จงบอกข้อแตกต่างระหว่างการเรียงอันดับทีละแถว (row-major order) และการเรียงอันดับทีละสดมภ์ (column-major order) Pascal ใช้วิธีใด
3. จงเขียนการประกาศชนิด สำหรับแถวลำดับซึ่งใช้เก็บประชากรหนอนปัจจุบัน (current worm population) ในกองซ้อนของกล่องใส่หนอน (worm boxes) ลักษณะกล่องเป็นรูปสี่เหลี่ยมฐาน 10 กล่อง X 7 กล่อง และแต่ละกองซ้อนมี 5 ชั้น
4. จงเขียนส่วนของโปรแกรมเพื่อแสดงผลผลบวกของค่า (ชนิด Real) ในแต่ละสดมภ์ ของแถวลำดับสองมิติชื่อ Table ที่มีแบบชนิดข้อมูลเป็น array [1 . . 5, 1 . . 3] of Real จะมีผลบวกสดมภ์ที่ชุดที่จะแสดงผล และในแต่ละสดมภ์มีสมาชิกแถวลำดับกี่ตัว

คำถามทบทวน

1. จงประกาศแถวลำดับ ซึ่งสามารถใช้เก็บชื่อเรื่องแต่ละชื่อของ Top40 ที่ติดอันดับสำหรับแต่ละสัปดาห์ของปี กำหนดให้ใช้ชื่อ TitleLength เป็นอักขระ 20 ตัว
2. จงเขียนการประกาศแถวลำดับชื่อ YearlyHours เพื่อเก็บจำนวนชั่วโมงทำงานของพนักงานแต่ละคน มีพนักงานห้าคนในแต่ละวันของสัปดาห์ แต่ละสัปดาห์ในหนึ่งปี
3. จงเขียนกระบวนการเพื่อคำนวณและแสดงผลจำนวนชั่วโมงทำงานทั้งหมดของพนักงานแต่ละคนบนวันที่กำหนดให้ (เป็นพารามิเตอร์อินพุต)
4. จงเขียนกระบวนการเพื่อคำนวณและแสดงผลจำนวนชั่วโมงทำงานทั้งหมดของพนักงานที่กำหนดให้ (เป็นพารามิเตอร์อินพุต)
5. จงเขียนการประกาศสำหรับแถวลำดับ CPUArray ซึ่งจะเก็บสมาชิก 20 ตัวของชนิด CPU เมื่อ CPU เป็นชนิดแถวลำดับ ซึ่งประกอบด้วยหน่วยเก็บสำหรับหน่วยข้อมูลชนิด Real ห้าตัว

เขียนโปรแกรม (Programming Projects)

1. จงเขียนชุดของกระบวนการเพื่อจัดดำเนินการของเมทริกซ์หนึ่งคู่ จัดหากระบวนการสำหรับการบวก การลบ และการคูณ กระบวนการแต่ละชุดควรจะตรวจสอบความสมเหตุสมผล (validate) พารามิเตอร์อินพุตของมัน (ตัวอย่างเช่น ตรวจสอบมิติของเมทริกซ์ทั้งหมด) ก่อนกระทำการจัดดำเนินการข้อมูลตามที่ต้องการ

2. จงเขียนโปรแกรมซึ่งใช้แถวลำดับสองมิติเพื่อเก็บผลลัพธ์ของการเลือกตั้ง
โปรแกรมควรจะเรียกกระบวนการงานต่อไปนี้ พารามิเตอร์ Tally เก็บผลลัพธ์ของการเลือกตั้ง

Procedure Initialize (var Tally) : อ่านคะแนนเลือกตั้งจำแนกตามรอบ (precint)
สำหรับผู้สมัครแต่ละคน

Procedure WriteTable (Tally) : แสดงผลผลลัพธ์การเลือกตั้งในรูปแบบตาราง
ข้างล่างนี้

Procedure CountBallot (var Tally) : ให้นำคะแนนเลือกตั้งที่ได้รับของผู้สมัคร
แต่ละคน และจำนวนร้อยละที่สมนัยกันของคะแนนเลือกตั้งทั้งหมด

Procedure FindWinner (Tally, var Cand1, var Cand2, var Over50) :
กำหนดตัวอักษรของผู้สมัครได้คะแนนสูงสุด (Cand1) และตัวอักษรของผู้สมัคร
ได้คะแนนสูงสุดอันดับสอง (Cand2) และตั้งให้ Over50 เป็นจริง ถ้าผู้สมัครซึ่งได้คะแนน
สูงสุดได้คะแนนเลือกตั้งมากกว่า 50%

Procedure DisplayWinner (Cand1, Cand2, Over50) :
แสดงชื่อของ Cand1 และข้อความ Won the election ถ้า Over50 เป็นจริง
กรณีอื่นๆ แสดงข้อความ

Runoff election between 2 highest-scoring candidates และ Cand1 และ
Cand2

จงเขียนโปรแกรมหลักซึ่งเรียกกระบวนการเหล่านี้ และวิ่งโปรแกรมสำหรับข้อมูล
ในตารางต่อไปนี้ และเมื่อผู้สมัคร C ได้คะแนนเพียง 108 เสียงเท่านั้นในการเลือกตั้งรอบที่
4 (precint 4)

Precint	CandidateA	CandidateB	CandidateC	CandidateD
1	192	48	206	37
2	147	90	312	21
3	186	12	121	38
4	114	21	408	39
5	267	13	382	29

3. จดัดแปร (modify) แบบฝึกหัดข้อ 2 ให้เป็นโปรแกรมทำงานด้วยรายการเลือก (menu-driven program) เชิงโต้ตอบ

เมนูตัวเลือก ควรเริ่มต้นตารางคะแนนเลือกตั้ง (แจ้งผู้ใช้ถึงจำนวนผู้สมัคร ชื่อผู้สมัคร และจำนวนรอบของการลงคะแนน)

แสดงตารางคะแนนเลือกตั้งด้วยจำนวนแถวและสดมภ์ทั้งหมด แสดงร้อยละของรอบทั้งหมด แสดงชื่อของผู้สมัครและคะแนนเลือกตั้งที่ผู้สมัครได้รับ (นับคะแนนและร้อยละของคะแนนเลือกตั้ง) แสดงชื่อของผู้ชนะการเลือกตั้ง (หรือชื่อต่างๆ ในเหตุการณ์ที่คะแนนเท่ากัน) และออกจาก (exit) โปรแกรม