

Multi-Dimensional Arrays In Pascal

In this section of notes you will learn about how to use multi-dimensional arrays (declaration, assignment and accessing data) as well as when to use them.

When To Use Arrays Of Different Dimensions

Determined by the data – the number of categories of information determines the no. of dimensions to use.

Examples:

- (1D array)
 - Tracking grades for a class
 - Each cell contains the grade for a student i.e., `grades[i]`
 - There is one dimension that specifies the student

↓ One dimension (which student)
- (2D array)
 - Personal finances program
 - One dimension of information specifies the financial category (cash in or cash out).
 - The other dimension is used to specify the time

When To Use Arrays Of Different Dimensions (2)

- (2D array continued)

Time

Financial category

	January	February	March	...
Income				
-Rent				
-Food				
-Fun				
-Transport				
-Misc				
Net income				

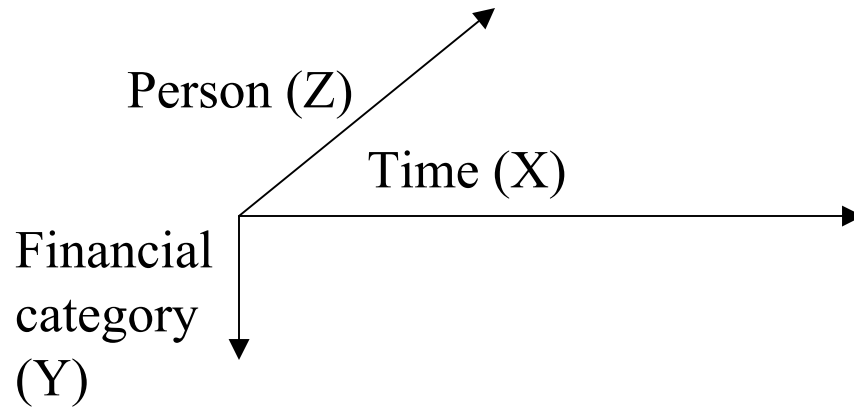
When To Use Arrays Of Different Dimensions (3)

- (2D array continued)
- Notice that each row is merely a 1D array
- (A 2D array is an array containing rows of 1D arrays)

	[1]	[2]	[3]	[4]
[1] Income				
[2] -Rent				
[3] -Food				
[4] -Fun				
[5] -Transport				
[6] -Misc				
[7] Net income				

When To Use Arrays Of Different Dimensions (4)

- (3D array – take the 2D array but allow for multiple people)
- The third dimension specifies whose finances are being tracked.



When To Use Arrays Of Different Dimensions (5)

	January	February	March	...
Income				
-Rent				
-Food				
-Fun				
-Transport				
-Misc				
Net income				

Declaring Multi-Dimensional Arrays

Syntax:

(Two dimensional arrays)

Name : array [*min..max*, *min..max*] of *type*;

(Three dimensional arrays)

Name : array [*min..max*, *min..max*, *min..max*] of *type*;

Rows Columns



Example:

johnFinances : array [1..7, 1..7] of real;

cube : array[1..3, 1..4, 1..6] of char;

Accessing / Assigning Values To Elements

Syntax:

```
name [row][column] := name [row][column];
```

Example:

```
finances [1][1] := 4500;  
writeln (finances[1][1]);
```


Example Program: Map Generator And Editor

You can find the full program in Unix under:

`/home/231/examples/array/map.p`

program map (input, output);

const

MAXROWS = 10;

MAXCOLUMNS = 10;

var

world : array[1..MAXROWS, 1..MAXCOLUMNS] of char;

r, c : integer;

randomValue : real;

quitChoice : char;

editChoice : char;

rowToEdit : integer;

columnToEdit : integer;

charToChange : char;

Example Program: Map Generator And Editor (2)

```
begin
for c := 1 to MAXCOLUMNS do
    world[1][c] := '-';
for c := 1 to MAXCOLUMNS do
    world[MAXROWS][c] := '-';
for r := 1 to MAXROWS do
    world[r][1] := '|';
for r := 1 to MAXROWS do
    world[r][MAXCOLUMNS] := '|';
```

Example Program: Map Generator And Editor (3)

```
for r := 2 to (MAXROWS-1) do
  begin
    for c:= 2 to (MAXCOLUMNS-1) do
      begin
        randomValue := Random;
        if (randomValue <= 0.05) then
          world [r][c] := '~'
        else if (randomValue <= 0.25) then
          world [r][c] := '^'
        else if (randomValue <= 0.30) then
          world [r][c] := 'C'
        else if (randomValue <= 0.40) then
          world [r][c] := 'T'
        else
          world [r][c] := ' ';
        end; (* inner for *)
      end; (* outer for *)
    end;
  end;
end;
```

Example Program: Map Generator And Editor (4)

```
repeat
  begin
    for r := 1 to MAXROWS do
      begin
        for c := 1 to MAXCOLUMNS do
          begin
            write(world[r][c]);
          end; (* inner for loop *)
        writeln;
      end; (* for loop - displays world *)
    writeln;
  write('Enter "Y" or "y" if you wish to edit the world or the return ');
  write('key otherwise: ');
  readln(editChoice);
```

Example Program: Map Generator And Editor (5)

```
if (editChoice = 'Y') OR (editChoice = 'y') then
  begin
    writeln;
    write('Enter row (2 - 9) to edit: ');
    readln(rowToEdit);
    write('Enter column (2 - 9) to edit: ');
    readln(columnToEdit);
    if (rowToEdit < 2) OR (rowToEdit > (MAXROWS-1))
      OR (columnToEdit < 2) OR
      (columnToEdit > (MAXCOLUMNS-1)) then
      writeln('Value for row must be in the range of 2 - 9')
```

Example Program: Map Generator And Editor (6)

```
else
  begin
    writeln('What do wish to change this square to? Choices include:');
    writeln('~" for water');
    writeln("^" for trees');
    writeln("C" for a city');
    writeln("T" for a town');
    writeln(" " (A space) for an open field');
    write('Enter choice and hit return: ');
    readln(charToChange);
    world[rowToEdit][columnToEdit] := charToChange;
  end; (* else *)
end; (* if edit mode chosen. *)
writeln('Type "Q" or "q" to quit, or return to continue: ');
readln(quitChoice);
end; (* repeat loop *)
until (quitChoice = 'Q') OR (quitChoice = 'q');
```

Example Program: Map Generator And Editor (7)

end.

Summary

You now know about multi-dimensional arrays:

- What is the syntax for declaring and using arrays of multiple dimensions?
- What determines the number of dimensions that an array should have?

Operations on 2D arrays:

- Initialization of elements
- Scanning selected parts of the array using loops