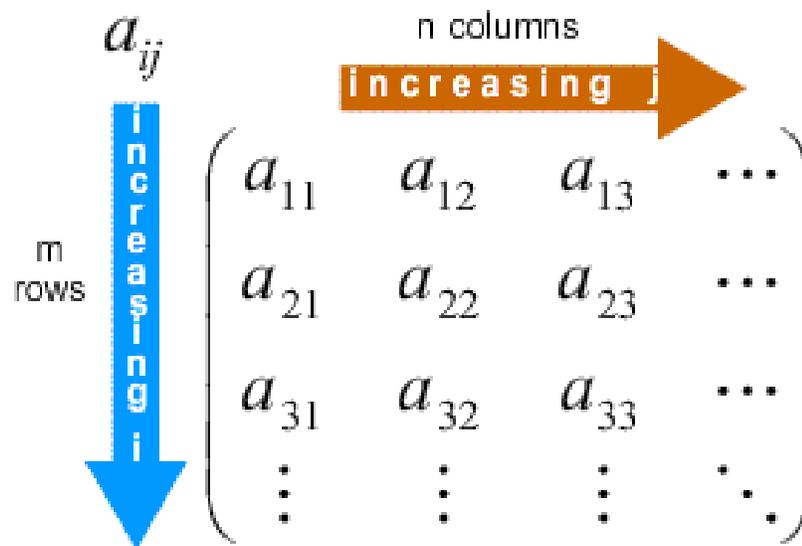


Grade 11

Worksheets

“Arrays – Worksheets No.2

Two Dimensional Arrays”



2018-2019

2D Arrays

Traditionally, an array is a static data structure. This means the array is declared with a specified number of elements of one specified data type and this does not change after compilation. However, many programming languages now allow an array to be dynamic. This means the array can grow in size if required.

In 2D arrays, the number of rows is given first, then the number of columns.

In **Pseudocode**, when we declare we have to specify the lower and upper bounds for each dimension.

In **Python**, all elements (rows and columns) are numbered from 0 number all elements from 0.

Pseudocode - 2D Array declaration

Pseudocode declaration examples:

```
DECLARE <identifier> : ARRAY [<lBound1>:<uBound1>,<lBound2>:<uBound2>) OF <dataType>
```

Where:

lBound1 : lower bound for the rows

uBound1 : upper bound for the rows

lBound2 : lower bound for the columns

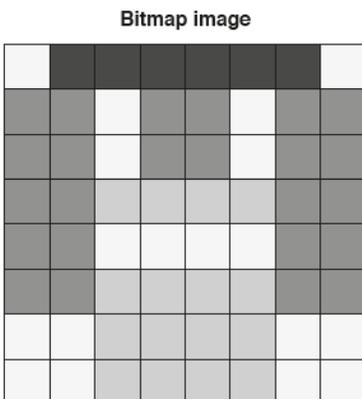
uBound2 : upper bound for the columns

Example - 2D Array declaration

Consider A 2D array, `Picture`, that contains data representing a bitmap image. Each element of the array represents one pixel of the image.

The image is grey-scale encoded where the value of each pixel ranges from 0 (representing black) to 255 (representing white) with intermediate values representing different levels of grey.

The following is an example of an image and the corresponding data values for the `Picture` array.



Values

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 240 | 10 | 10 | 10 | 10 | 10 | 10 | 240 |
| 80 | 80 | 240 | 80 | 80 | 240 | 80 | 80 |
| 80 | 80 | 240 | 80 | 80 | 240 | 80 | 80 |
| 80 | 80 | 150 | 150 | 150 | 150 | 80 | 80 |
| 80 | 80 | 240 | 240 | 240 | 240 | 80 | 80 |
| 80 | 80 | 150 | 150 | 150 | 150 | 80 | 80 |
| 240 | 240 | 150 | 150 | 150 | 150 | 240 | 240 |
| 240 | 240 | 150 | 150 | 150 | 150 | 240 | 240 |

In pseudocode, the `Picture` array is declared as follows:

```
DECLARE Picture : ARRAY[1:8, 1:8] OF INTEGER
```

Pseudocode - Accessing 2D arrays

A specific element in a 2D Array (table) is accessed using an index pair. In pseudocode this is written as:

```
Picture[5,4] = 0 // sets the element in row 5 and column 4 to zero
```

Python - Syntax definitions

In Python, there are no arrays. The equivalent to a 2D array data structure is a list of lists. As there are no declarations, the only way to generate a list is to initialize one.

| | | |
|---------------|---|--|
| Python | <pre>Board = [[0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0]] Board = [[0 for i in range(7)] for j in range(6)] Board = [[0] * 7] * 6</pre> | <p>2D lists can be initialised in a similar way to 1D lists. Remember that elements are numbered from 0.</p> <p>These are alternative ways of initialising a 6 × 7 list. The rows are numbered 0 to 5 and the columns 0 to 6.</p> <p>The upper value of the range is not included.</p> |
|---------------|---|--|

Python - Accessing 2D arrays

| | | |
|---------------|----------------------------|--|
| Python | <pre>Board[2][3] = 0</pre> | Elements are numbered from 0 in Python, so [3] gives access to the fourth element. |
|---------------|----------------------------|--|

1) Design an algorithm (Pseudocode or Python) to create the following 2D arrays.

a)

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

b)

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |

2) For the array mentioned above in the “2D Array declaration Example”:

- a) Create a function `AddRowCells()` that will calculate the sum of the values of the first row.
- b) Create a function `AddColumnCells()` that will calculate the sum of the values of the last column.
- c) Create a function `IncreaseValue()` that will increase the values of each cell by 50

3) For the array mentioned above in the “2D Array declaration Example”: A function, `Lighten()`, is required to lighten the image. Lightening an image may cause it to ‘burn out’. An image is said to be ‘burnt out’ if any pixel is set to the maximum value of 255. The function `Lighten()` will:

- increase the value of each pixel by 10%
 - Return `TRUE` if the resultant image is ‘burnt out’.
- a) Write **pseudocode or Python code** to implement the `Lighten()` function.

4) A function, `ProcessMarks()`, is required to analyse test marks for a class of students.

- There are 20 students in the class.
- A mark is between 0 and 100.
- The marks for the class are stored in an array, `Mark`, which has 20 elements.
- The array is passed to the function as a parameter.
- The function will output a message stating the average and highest marks. For example:
"The average mark is 34 and the highest mark is 76"
- The function returns the subscript of the highest mark.
 - a) Write **program code** to implement the `ProcessMarks()` function.

5) Create a program in Python that uses the `random.randint()` function to initialize a 5 rows, 5 columns 2D array with random integer values between 1 and 100. Your program should (after the array is initialized) also print the array elements.

6) Design an algorithm (Pseudocode or Python) that will go through the array you created in exercise 5, find the largest value and display this value along with coordinates of the cells where it was located.

7) Create a program in Python that uses the `random.randint()` function to initialize a 5 rows, 5 columns 2D array with random integer values between 1 and 100. For this Array :

- a) Create a function that calculates and displays the average of each row.
- b) Create a function that calculates and displays the average of each column.

8) A 6 row 1D array `Names[0:4]` is used to hold the name of the students of a class. A 2D array `Grades` holds the corresponding grades of each student on 3 assessments. Design an algorithm (Pseudocode or Python) that creates the arrays `Names`, `Grades`, calculates the assessment grade average for each student and stores it in a 1D array `FinalGrade[0:5]`. Your program should find and display the name of the student(s) with the highest average grade.

9) Design an algorithm in Pseudocode or Python code which will take as input two integer numbers "row" and "column" and generate a 2D array where the value of each cell is product of the corresponding row and column.

10) Design an algorithm (Pseudocode or Python) to create the following 2D arrays.

a)

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 |

b)

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 |

c)

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

d)

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |