

INTRODUCTION TO VBA PROGRAMMING

LESSON6

dario.bonino@polito.it



Agenda

- Arrays
- Matrices

Arrays

Arrays

- Allow to store vectorial data
 - Geometric vectors
 - Sets of data having something in common
 - ...
- Declared as
 - **Dim** *array_name* (*begin To end*) **As**
Type

Arrays

- Array elements are accessed by means of a position index
 - ▣ `array(1)`
 - Gets the value stored inside the array variable at position 1
- *Begin*
 - ▣ The index of the first element in the array
 - ▣ 0 if not specified
- *End*
 - ▣ The index of the last element in the array

Arrays

□ Examples

▣ Declaration

- `Dim array1 (1 To 10) As Integer`
- `Dim array2 (20) As Single`
- `Dim array3 (3 To 40) As String`

▣ Usage

- `Sum = array(1) + array(4)`
- `Username = array3(3)`

Example

- Write a program that asks ten values to the user and shows them in the reverse order

```
Dim values (9) As Integer
Dim i As Integer
For i = 0 To 9
    values(i) = InputBox("Insert an integer number")
Next i
` show in the reverse order
For i = 9 To 0 Step -1
    MsgBox(values(i))
Next i
```

Array functions

- **ReDim** *array_name*(*newdimension*)
 - ▣ Sizes a given array to *newdimension* (usually provided by a variable)
 - ▣ The space required for holding values is dynamically allocated
 - ▣ Example:
 - Dim array () As Integer
 - Size = InputBox("How many values?")
 - ReDim array(1 To Size)
- **Erase** *array_variable*
 - ▣ Empties the *array_variable* if it was declared as "static"
 - If Integer, all values are set to 0
 - If String, all values are set to ""
 - ▣ Deletes the array if it was declared through the ReDim function

Exercise 1

- Write a program that asks to the user a number of values to insert. Given the number of values, the program asks all such values, one by one. Finally it provides back the values ordered from the smallest to the biggest
- Suggestion: use 2 arrays, one for storing the user-inserted values and one for holding the ordered list.

Exercise 2

- A data management program provides access to certain functionalities by checking the user identity through a username and a password. Supposing that allowed usernames and passwords are stored in 2 arrays with usernames and associated passwords in the same position, write a program that asks the username and the password and verifies if the user is valid or not
- Suggestion:
 - Username() = {user1,user2,user3,...}
 - Password() = {pwd1, pwd2, pwd3,...}

Exercise 3

- Write a program that asks the user for the values of two arrays. The values have to be specified as a single string where values are separated by a comma
 - ▣ Array1 → 12 , 34, 5, 67, 98, 33
- The program should list the values that occur in both arrays

Matrices

Matrices

- Represent bi-dimensional structures like
 - ▣ Tables
 - ▣ Mathematical matrices
- Declared as
 - ▣ `Dim matrix (0 To 9, 0 To 9) As Integer`
- Values are accessed by means of two indexes
 - ▣ `cell = matrix(3,8)`
 - The cell located on the 3^o row and on the 8^o column

Matrices functions

- **ReDim** *array_name* (*begin To end*,
begin To end)
 - ▣ To dynamically allocate a matrix

Example

□ Matrix scan

- ▣ We want to print out all the values of a matrix `Mat` having a 8x8 size

```
Dim out As String
```

```
....
```

```
For i = 1 To 8
```

```
    out = out & "|"
```

```
    For j= 1 To 8
```

```
        out = out & Mat(i,j) & "|"
```

```
    Next j
```

```
    out = out & vbNewLine
```

```
Next i
```

```
MsgBox out
```

Exercise 4

- Write a program that asks to the user a number between 1 and 20 and automatically generates the multiplication table of order equal to the number.
- Then, the program must challenge the user to answer questions about multiplications
 - ▣ It randomly selects two numbers between 1 and the multiplication table size
 - ▣ It asks the result of the multiplication of such numbers to the user and checks provided input against the value stored in the multiplication table
 - ▣ If correct the program shall write “right” otherwise “wrong”
- Challenges must continue until the user writes “stop”

Exercise 5

- Write a program that given a matrix M computes the smoothed matrix MS using the following algorithm
 - ▣ The $MS(i,j)$ value is provided by the average of the values of the cells surrounding the same position in M
 - $MS(i, j) = (M(i-1, j-1) + M(i-1, j) + M(i-1, j+1) + M(i, j-1) + M(i, j) + M(i, j+1) + M(i+1, j-1) + M(i+1, j) + M(i+1, j+1)) / 9$