INTRODUCTION TO VBA PROGRAMMING

LESSON7

dario.bonino@polito.it

cc BY NC SA

e-Lite

Agenda

- Files
- Output to file
- Input from file

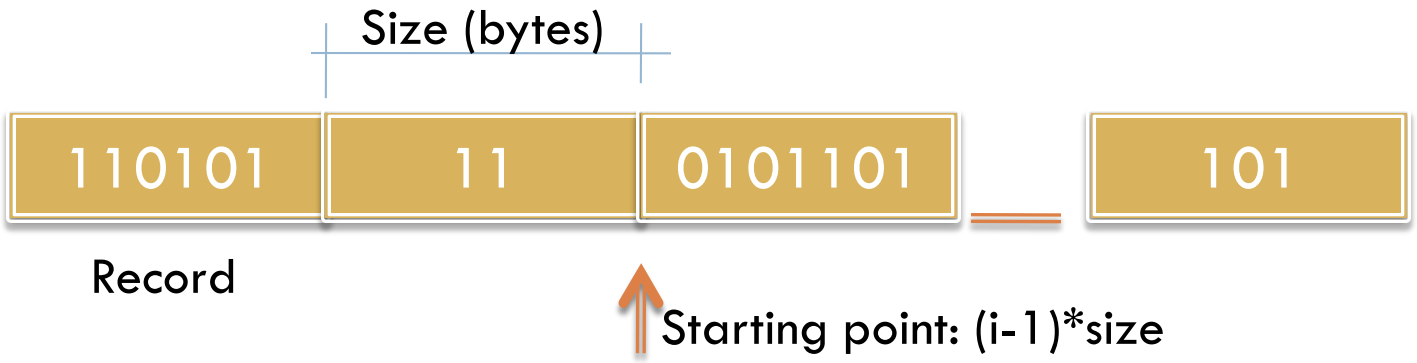
Files

Files

- Information (bytes) stored on a “mass storage device”
  - E.g. harddrive., memory cards, diskettes, usb sticks
- 2 main types
  - Sequential
    - Data is written as a sequence of “records”
    - To access the i-th record all previous records must be read (that’s why is called sequential)
    - Differently sized records depending on the stored types

Files

- Random Access
  - All records have the same size
    - If the size is “greater” than the size of the stored type, “holes” are left
    - Records can be directly accessed on the basis of their position in the file (they all have the same size)



Opening a file

- To open a file
  - **Open** *filename* **For** *mode* **As** *#number*
  - *Filename*
    - The name of the file to open
  - *Mode*
    - How the file has to be opened
      - Input – to read the file
      - Output – to write the file
      - Append – to append content at the end of the file
  - *#number*
    - An identification number for the file

Closing a file

- To close a file
  - The file numeric identifier is needed
    - `Close #fileid1`
  - Multiple files can be closed at the same time
    - `Close #fileid1, #fileid2, #fileid3`
  - To close all files
    - `Close`

Writing to files

Writing to file

- Print allows to print some expressions to a file
  - `Print #fileid, expression1; expression2; expression3`
  - `#fileid` – identifies the file to which expressions must be output
  - `expression` – what to print in the file
    - Strings
    - Result of numerical expressions
    - …
  - `Print #3, “The result is: ”; result`

Example 1

```
Open “results.txt” For Output As #4
Print #4, “#results”
Print #4,”|x|y|”
For i=0 To 10
  Print #4, “|”;x(i);”|”;y(i);”|”
Next I
Close #4
```

Exercise 1

- Write a program that given a number N computes all the powers of 2 from 0 to N
  - $2^0=1$, $2^1=2$,..... $2^N=?$
- The computed values must be saved on a file named powers.txt with a tab separated format
  - 0 1
  - 1 2
  - 2 4
  - 3 8

Reading from files

Reading from files

- To read one or more values from a file
  - **`Input`** *`#fileid, variable1, variable2`*
  - *`#fileid`*
    - The numeric identifier of the file to be read
  - *`variable1…n`*
    - The variable in which to store the values read from the file
    - The filling behavior changes depending on the variable type

Reading from files

- If the variable to be read is a string, the input function
  - skips any initial white space (spaces and TABs);
  - reads all the characters until a string field separator (commas and new-line characters) is found;
  - assigns them to variable.

Reading from files

- If the variable to be read is numeric, the input function
  - skips any initial white space (spaces and TABs);
  - reads all the contiguous characters
  - passes them to the `Val` function in order to have a numerical value (if a reading returns a non-number or an empty string, value 0 is produced);
  - reads all characters until a number field separator (commas, new-line characters, spaces, and TABs) is found (after a field, white space before a comma or new-line is ignored);
  - assigns the resulting value to the variable.

Example

- Suppose we want to read
  - one two, three , four five

    Six, seven , eight nine, ten
- We write the following statements
  - `Input #1, A`
  - `Input #1, B`
  - `Input #1, C`
  - `Input #1, D`
  - `Input #1, E`
  - `Input #1, F`
  - `Input #1, G`

Example (cont.d)

- What we expect to be the value of ABCDEFG?
  - A = “one two”
  - B = “three”
  - C = “four five”
  - D = “Six”
  - E = “seven”
  - F = “eight nine”
  - G = “ten”

Example2

- August 16, 2006 Wednesday

```
Dim A As String
Dim B As Integer
Dim C As String
Input #1, A, B, C
```

- A = “August 16”
- B = 2006
- C = Wednesday

Example3

- Characters enclosed in quotes are read as single strings
  - “Hello Everybody”
  - `Input #1, A`
  - `A = “Hello Everybody”`

Utilities

- The `write` function works as the `print` function but already encloses string between double quotes and writes the correct field separator characters
  - `Write #1, variable1, variable2,…, variableN`
- To read one entire line inside a variable the Line Input instruction can be used
  - `Line Input #1, theWholeLine`

End of File?

- We can read from files but…
  - How can we detect the file end?
- The EOF (End Of File) function allows to check if a file still contains something to read
  - True – the end of the file has been reached
  - False – the file has some more data to read
  - `EOF(#fileId)`
  - Usually we cannot assume to know the number of fields to be read in a file….

Example

- Read a file line by line and show the line content until the file is finished

```
Open “input.txt” For Input As #1
Do While Not EOF(1)
   Line Input #1, line
   MsgBox (line)
Loop
```

Example 2

- Use a text editor and create a file with some numbers, one on each line. Now write a program that asks the user for the name of the file to read, then calculates and displays how many values there are in that file, their maximum and minimum value, their sum and average.

Exercise 2

- Write a program that reads all the lines of a file, and writes them in reverse order (i.e. printing the line content from right to left) in another file.
- Display a MsgBox reporting how many rows have been copied.

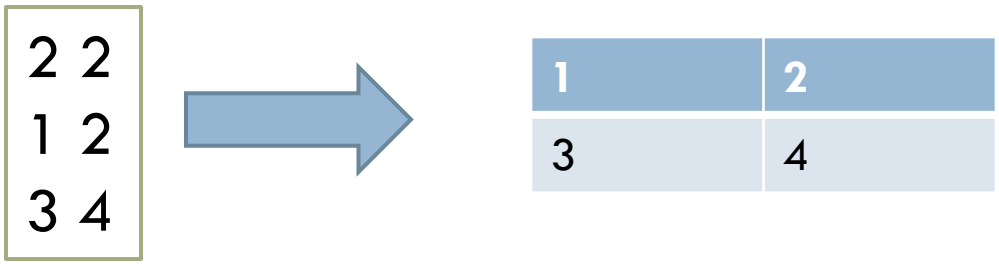
Exercise 3

- Write a program to encrypt and decrypt a text file by using Julius Caesar's cipher method. The program asks the user for the input file, the output file and a secret key (a whole number). Encryption is achieved by substituting uppercase and lowercase letters with other letters based on the *key. Each letter position is shifted by key positions: for example, suppose key=3, then "A" becomes "D", "B" becomes "E", etc. and "X" becomes "A", "Y" becomes "B", and "Z" becomes "C". The same for lowercase letters. To decrypt an encrypted text, a negative value for key (e.g. –3) is used.*

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Exercise 4

- Write a program that reads a matrix from a file
  - The file reports the values of the matrix cells on separated rows, one per each matrix row
  - Values inside a row are separated by a space
  - The first line of the file contains 2 values respectively reporting the number of rows and columns of the matrix



- Display the matrix in a message box

Exercise 5

- Write a program that reads a matrix in the format defined in Exercise 4 and writes a new file where the rows and the cells of the matrix are exchanged (transposed matrix)

1	2
3	4

1	3
2	4

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
  - The initial matrix must be read from a file having the format defined in Exercise 4
  - The smoothed matrix must be saved in a file having the same format