

## File/directory processing

Write a Java, C# or C/C++ program (the choice is yours) for file/directory processing according to the following rules. The program requested for this project must have a text menu like this:

- 0 – Exit
  - 1 – Select directory
  - 2 – List directory content (first level)
  - 3 – List directory content (all levels)
  - 4 – Delete file
  - 5 – Display file (hexadecimal view)
  - 6 – Encrypt file (XOR with password)
  - 7 – Decrypt file (XOR with password)
- Select option:

The menu is displayed and the user must select an option (a number between 0 and 7). The action corresponding to the selection is performed, then the menu is displayed again and the user can choose another option. This cycle is repeated until the user selects 0, which exits the loop and ends the program.

The options are:

- 0 – Exit

This options ends the program

- 1 – Select directory

The user is prompted for a directory [absolute] name. This is the first options that must be selected by the user. All the options below are working on the directory selected here. After performing several operations on the selected directory, the user can select another directory and work with it.

- 2 – List directory content (first level)

This option displays the content of the selected directory on the screen. All the files and sub-directories from the first level must be displayed (files and directories should be listed separately). If no directory was selected an error message must be displayed.

- 3 – List directory content (all levels)

This option displays the content of the selected directory on the screen. All the files and sub-directories from the first and subsequent levels must be displayed (files and directories should be listed separately). If no directory was selected an error message must be displayed.

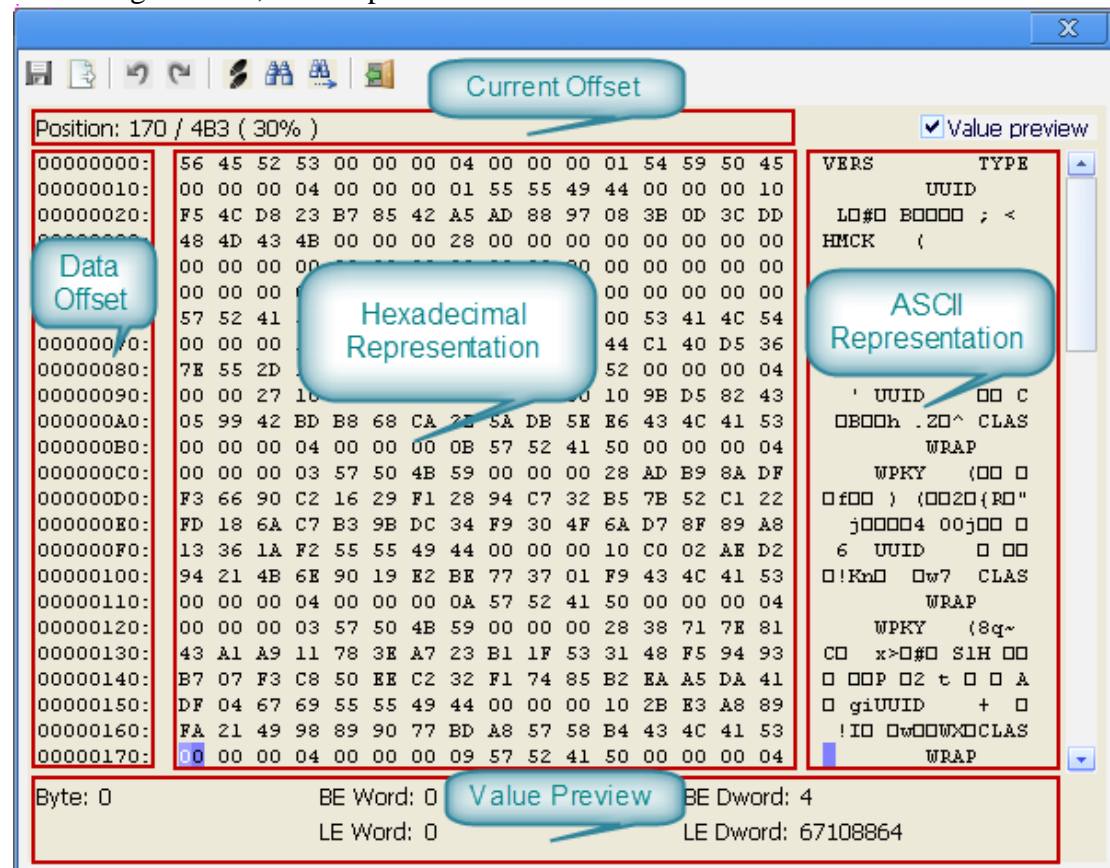
- 4 – Delete file

This option prompts the user for a filename and deletes that file from the selected directory. If no directory was selected an error message must be displayed. If the directory does not contain the file specified by the user, an error message must be displayed. The filename does not include any path, it's just the name of the file.

- 5 – Display file (hexadecimal view)

This option prompts the user for a filename (from the selected directory) and displays the content of that file on the screen, in hexadecimal view. If no directory was selected an error message must be displayed. If the directory does not contain the file specified by the user, an error message must be displayed. The filename does not include any path, it's just the name of the file.

Note 1: The hexadecimal view displays each byte of the file in hexadecimal; it should look something like this, but simpler:



Note2: The ASCII representation, the Current Offset and the Value Preview are NOT required; also there is no window, no graphics, no colors, just black and white text display. Being a text only display, you are required to display ONLY the Data Offset and the Hexadecimal Representation.

## 6 – Encrypt file (XOR with password)

This option prompts the user for a password (max 256 bytes long, may contain letters, digits, other characters) and then prompts the user for a filename and encrypts the content of the selected file using that password. The encryption method is very simple: just XOR the password with the file content byte after byte; the password being shorter than the file content, you must repeat the password as needed.

Example:

```
passwordpasswordpasswordpasswordpasswordpasswordpass
this is the file content that we need to encrypt now
-----
chiphertext is obtained here by XORing byte to byte
```

Note1: the user must be prompted for the filename of the encrypted file (containing the ciphertext) as well, otherwise we would need to overwrite the initial file.

Note2: If no directory was selected an error message must be displayed. If the directory does not contain either of the files specified by the user, an error message must be displayed. The filenames do not include any path.

#### 7 – Decrypt file (XOR with password)

This option prompts the user for a password (max 256 bytes long, may contain letters, digits, other characters) and then prompts the user for a filename and decrypts the content of the selected file using that password. The decryption method is very simple: just XOR the password with the file content byte after byte; the password being shorter than the file content, you must repeat the password as needed.

Example:

```
passwordpasswordpasswordpasswordpasswordpasswordpass  
chiphertext is here ...  
-----  
this is the file content that we had initially
```

Note1: the user must be prompted for the filename of the decrypted file as well, otherwise we would need to overwrite the initial file.

Note2: it may seem strange that we are using the same operation (XOR) both for encryption and for decryption, but this is how XOR (exclusive OR) works.

Note3: If no directory was selected an error message must be displayed. If the directory does not contain either of the files specified by the user, an error message must be displayed. The filenames do not include any path.

Testing:

1. You should use this file as the test file for Display, Encryption and Decryption.
2. For encryption, use the following password “Qwertyuiop[123\$4\$567]”
3. After encrypting this file with the above password, you obtain an encrypted file of the same length; after decrypting that file with the same password you should obtain an exact replica of this file, and this must be reflected in the Report document by appropriate screenshots showing the content of the files involved in these operations.

Deliverables:

1. Source code (zipped)
2. A Report showing the program performing each of the 7 actions correctly (use screenshots), plus screenshots showing the error reporting situations.