Cairo University, Faculty of Computers
and Artificial Intelligence

FACULTY OF COMPUTERS AND ARTIFICIAL INTELLIGENCE, CAIRO UNIVERSITY

CS112: Structured Programming
Year 2021 - 2022

Second Semester

Assignment 4 – Text Editor Description

Version 1.0

Deliver ONE fully integrated program with all parts

Course Instructors:

Dr. Mohammad El-Ramly

Revision History

Detailed Description of the Text Editor Program

It is required to develop a Text Editor program that displays a menu with the list of options show in the table below. At the start:

1- The user enters the name of the file he wants to deal with

2- The program will test if such a file exists or not. If it does, it will display a message: "**This File Already Exists**" and it will open it for read and write. If it does not exist, it will create a file with the given name and will print a message: "**This is a new file. I created it for you ☺**"

3- After that the following menu will be displayed. The user chooses an option to apply and after executing it, the program re-displays the menu till Exit is chosen.

English	Arabic
1. Add new text to the end of the file	١. إضافة المزيد من المحتوى لنهاية الملف
2. Display the content of the file	٢. عرض محتويات الملف
3. Empty the file	٣. مسح محتويات الملف
4. Encrypt the file content	٤. تشفير الملف
5. Decrypt the file content	٥. فك شفرة الملف
6. Merge another file	٦. دمج الملف مع ملف آخر
7. Count the number of words in the file.	٧. عد الكلمات الموجودة بالملف
8. Count the number of characters in the file	٨. عد الحروف الموجودة بالملف
9. Count the number of lines in the file	٩. عد السطور الموجودة بالملف
10. Search for a word in the file	١٠. البحث عن كلمة فى داخل الملف
11. Count the number of times a word exists in the file	١١. عد عدد مرات تكرار كلمة فى الملف
12. Turn the file content to upper case.	١٢. تحويل محتويات الملف للحروف الكبيرة
13. Turn the file content to lower case.	١٣. تحويل محتويات الملف للحروف الصغيرة
14. Turn file content to 1ˢᵗ caps (1st char of each word is capital)	١٤. تحويل محتويات الملف لأول حرف كبير و الباقى صغير
15. Save	١٥. خزن الملف
16. Exit	١٦. الخروج

4. The program should use **structured programming** and should be divided into functions.

5. The program should use separate compilation and be divided to header file, cpp library file, and main cpp program file.

6. The above functions are explained below:

1- **Adding content** functionality will open the file for writing in append mode (which means that the original content of the file is not lost but new the writing will be added to the end of the file) and will allow the user to type in text that is stored in the file, until he presses EOF character (Ctrl+z) which ends the writing and returns the user to the menu.

2- **Display content** functionality displays the current file content to the user and returns him to the menu.

3- **Empty the file** clears the file and erases its content and returns to the main menu.

4- **Encrypt the file content** will re-write the file content but shifted one ASCII character, i.e., each character will be replaced by the character following it in the ASCII table. The menu is re-displayed after the operation. For example, the following file:

--------------------------- **Original File** ----------------------------
Allah watches me. Allah sees me all the time.

Will be

-------------------------- **Encrypted File** --------------------------
Bmmbi!xbudifu!nf/!Bmmbi!tfft!nf!bii!uif!ujnf/

5- **Decrypting the file content** is the opposite of the pervious operation, i.e., it is re-writing the original content from the encrypted file. The menu is re-displayed after the operation.

6- **Merge another file** functionality will take from the user the name of another file. If this file does not exist, it will display a message "Invalid File Name". If the file exists, the program will add the given file to the end of the current file and returns to the main menu.

```
---------------------------------- Original File ------------------------------
Allah spreads His hand in the night for the sinner of the day to repent.
--------------------------------------------------------------------------------

------------------------- The New File to Merge-------------------------
And He spreads His hand in the daytime for the night sinner to repent.
--------------------------------------------------------------------------------

------------------------ Original File after Merge ----------------------
Allah spreads His hand in the night for the sinner of the day to repent. And He spreads His
hand in the daytime for the night sinner to repent.
--------------------------------------------------------------------------------
```

7- **Count the number of words** functionality prints the number of words in the file. A word is any sequence of characters not separated by white spaces (space, tab, new line, etc.)

8- **Count the number of characters** functionality prints the number of characters in the file and returns to the main menu.

9- **Count the number of lines** functionality prints the number of lines in the file and returns to the main menu.

10- **Search for a word in the file** functionality will take a word from the user and print either: "Word was not found in the file ☹" or "Word was found in the file ☺. The search should be case insensitive. So, the word "Ali", "ali" and "aLI" are the same.

11- **Count the number of times a word exists in the file** takes from the user a word and prints the number of times the word exists in the file as a separate word and returns to the main menu. For example:

```
---------------------------------- Original File ------------------------------
In-shaa-Allah, Allah will answer your duaa as ALLAH said: "And your Lord has said,
supplicate to me, and I will answer you.
--------------------------------------------------------------------------------
```

If the searched word is "Allah", The search will print: **The word "Allah" was found 2 times in the file.** Note that "In-shaa-Allah" was not counted.

The search should be case insensitive, so, if the words "Allah", "allah" and "AllaH" are all the same word.

12- **Turn the content to upper case** functionality will re-write the file all in upper case and returns to the main menu.

```
-------------------------- Original File --------------------------
The Messenger Said: Treat your sick with Sadakah (Charity).
----------------------------------------------------------------------
```

Will be

```
-------------------------- Uppercase File -------------------------
THE MESSENGER SAID: TREAT YOUR SICK WITH SADAKAH (CHARITY).
----------------------------------------------------------------------
```

13- **Turn the content to lower case** functionality will re-write the file all in lower case and returns to the main menu.

--------------------------- **Original File** ---------------------------
The Messenger Said: Treat your sick with Sadakah (Charity).

Will be

-------------------------- **Lowercase File** --------------------------
The messenger said: treat your sick with sadakah (charity).

14- **Turn file content to first caps** functionality will re-write the file all in lower case and except the first letter of each word will be capital and returns to the main menu.

--------------------------- **Original File** ---------------------------
smiling in THE FACE of your brother is a charity.

Will be

-------------------------- **First Caps File** --------------------------
Smiling In The Face Of Your Brother Is A Charity.

15- **Save** allows saving the file content again in the same file or under a different name.

16- **Exit** terminates the program:

Important Notes

1- **cctype** library has functions for character manipulation like **tolower** and **topper.**

2- **cin.get()** reads one character by one character from an input stream and can be used to the input one character at a time.

3- Operator **static_cast <...>** can be used to convert from a type to another, e.g. **static_cast <char> (x)** for **int x** gives the corresponding **char**.

4- Opening a file requires the name to be C-style string variable (array of characters) and not C++ **string**. So, you may need to use the function **c_str()** as follows:

```
ifstream inputFile;
string myFile;
cin >> myFile;
inputFile.open (myFile.c_str());
//myFile.c_str() returns the C-style string equivalent to myFile C++ string.
```

Example Run

```
Welcome to xxxxxxxxxx Text Editor   /// xxxxxxxx is your team name

Please Enter Your File Name: myFile.txt

File myFile.txt does not exist. I created it for you.
```

1. Add new text to the end of the file
2. Display the content of the file
3. Empty the file
4. Encrypt the file content

5. Decrypt the file content

……………………………………………………………………

16. Exit
```
Please Enter your choice ==> 1

Please type your text and finish by Ctrl+Z

I love C++ very much.

^Z
```

1. Add new text to the end of the file

…………………………………………….

16. Exit
```
Please Enter your choice ==> 4

File is Encrypted and secured
```

1. Add new text to the end of the file

…………………………………………….

16. Exit
```
Please Enter your choice ==> 2

-------------------------- MyFile.txt ------------------------------

J!mpwf!D,,!wfsz!nvdi/
```

1. Add new text to the end of the file

…………………………………………….

16. Exit
```
Please Enter your choice ==> 5

File is Decrypted
```

1. Add new text to the end of the file

…………………………………………….

16. Exit
```
Please Enter your choice ==> 2

-------------------------- MyFile.txt ------------------------------

I love C++ very much.
```

1. Add new text to the end of the file

…………………………………………….

16. Exit
```
Please Enter your choice ==> 15

Enter a new file name or press enter to save on the same file ==> (Enter)

File is saved.
```