



**National University**  
of computer and emerging sciences

## **DATA STRUCTURES PROJECT REPORT**

**PROJECT TITLE: TEXT EDITOR**

**COURSE INSTRUCTOR:**

Dr Fahad Sherwani

### **GROUP MEMBERS:**

- M. Hamood Siddiqui (21K-4539)
- M. Umar (21K-3219)
- Abdul Rehman (21K-4662)

## **ABSTRACT:**

The data structures we have studied have resulted in this creation of an effective everyday use console-based software program "Text-Editor". It is an ambitious tool which can be used to solve your problem like writing and reading text files.

## **INTRODUCTION:**

Text Editor is a console-based program that provides basic functionalities of a text editor including

- Creation of new files and adding text to it
- Accessing already created files and view or edit their content

The project provides different functionalities to the user while editing or viewing files which include

- Editing any line by entering the line number
- Adding text to any desired position in file
- Deleting any specific line by entering the line number
- Undo/redo functions
- Copy and move line functions
- Printing specific lines by entering line number

The interface has been tried to keep simple and unnecessary design elements are avoided. The user can interact with the program by entering specific commands.

## **METHODOLOGY:**

The project has been implemented using Stack, Queue and Linked List data structures. Custom implementation of Linked List and Stack has been used whereas built-in implementation of queue is used.

File stream and string functions have been used to create, view and add text to files. All the text data entered by the user is stored word by word into Queue and stores to the file. The editing functions make use of Stack and Linked List to perform operations on data.

## **STACK:**

Stack has been used for the undo and redo functions as Stack follows the LIFO (Last in First Out) principle. The LIFO Principle of Stack allows us to perform undo and redo operations on user entered data by pushing and popping out the data whenever needed.

## **LINKED LIST:**

The program makes use of linked list to perform different editing options including copy, paste, delete, insert at line functions. All the user entered text is inserted into linked list and then edit operations are performed by getting to the desired node and implementing the desired operation. The file is then recreating by accessing every linked list node containing a single line of text.

Functions for printing multiple lines or printing a single line at any desired position also make use the use of Linked List data structures using the same implementation discussed above.

## **OTHER UTILITIES:**

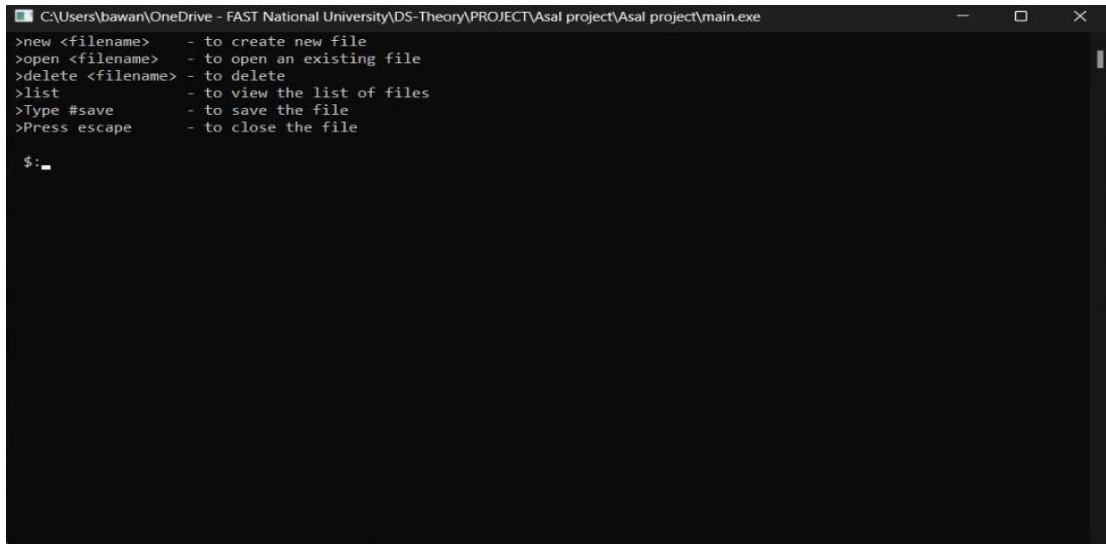
The program also makes use of some basic standard c Libraries to carry out string and file or display operations. All the commands that user uses during the running of a program are saved to a file and can be accessed using the up/down cursor for the ease of user.

## **CODE IMPLEMENTATION:**

the program is divided into different modules, so it is very easy to understand and add or change some code if needed. The custom implementation of linked list and stack data structures have been defined in "stack.h" and "linkedlist.h" header files. The menu is designed in the "menu.h" header file and the utility functions are defined in "util-funs.h" header file. all the functions that are heading start the program like creating a new file or viewing an new file or editing a new file are defined in this file but the actual implementation of the functions carrying out the low-level functionality of a text-editor are defined in "view-funs.h" and "edit-funs.h" header file. The "util-funs.h" header file includes the other two files and then based on the user input, performs the desired task by calling out the functions defined there.

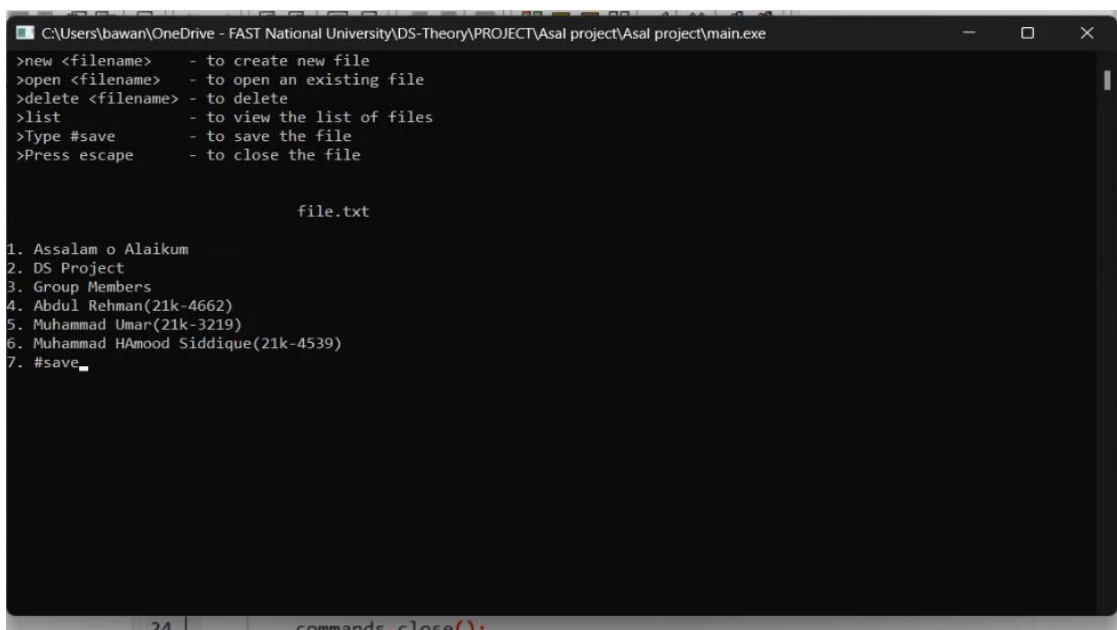
## RESULTS:

- main menu/creating new files:



```
C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe
>new <filename>      - to create new file
>open <filename>     - to open an existing file
>delete <filename>   - to delete
>list                - to view the list of files
>Type #save          - to save the file
>Press escape        - to close the file

$:
```



```
C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe
>new <filename>      - to create new file
>open <filename>     - to open an existing file
>delete <filename>   - to delete
>list                - to view the list of files
>Type #save          - to save the file
>Press escape        - to close the file

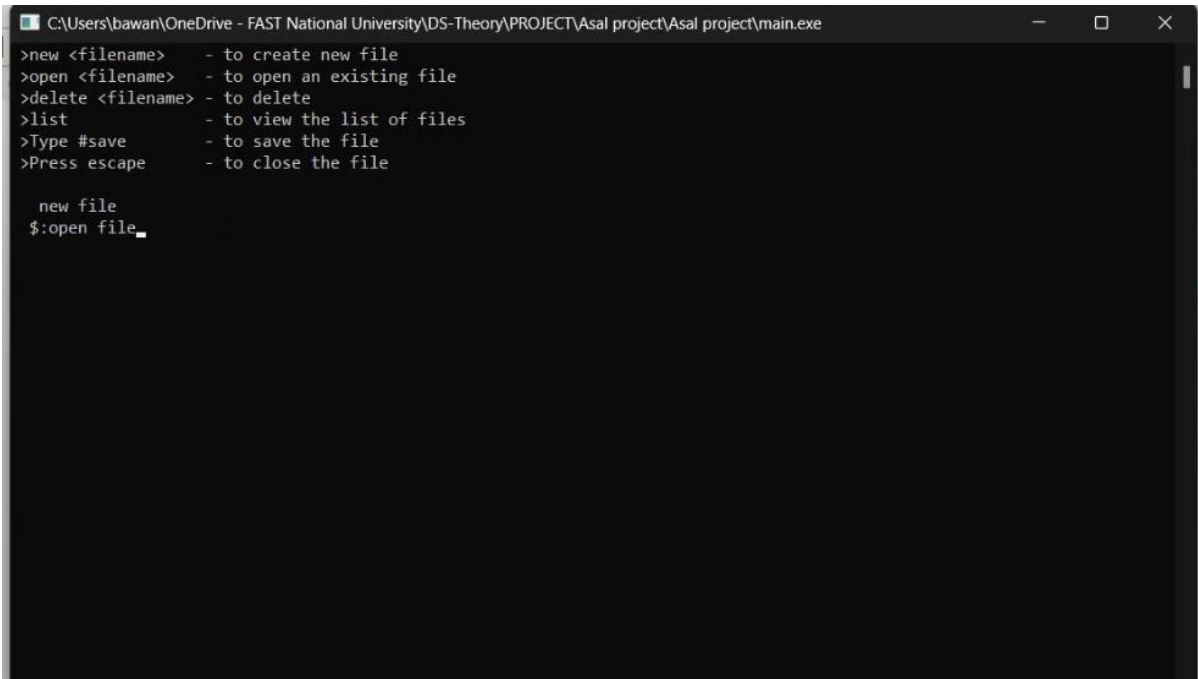
                                file.txt

1. Assalam o Alaikum
2. DS Project
3. Group Members
4. Abdul Rehman(21k-4662)
5. Muhammad Umar(21k-3219)
6. Muhammad HAMood Siddique(21k-4539)
7. #save

24 | commands.close();
```

user can simply interact with the program by entering commands. new command will create a new name-specified text file if it doesn't exist already else will throw an error whereas open and delete commands will respectively open and delete the name-specified text file. List command will list all the files created using the "new" keyword during the running memory of program. the user can simply save the file by using #save command after entering all the desired text.

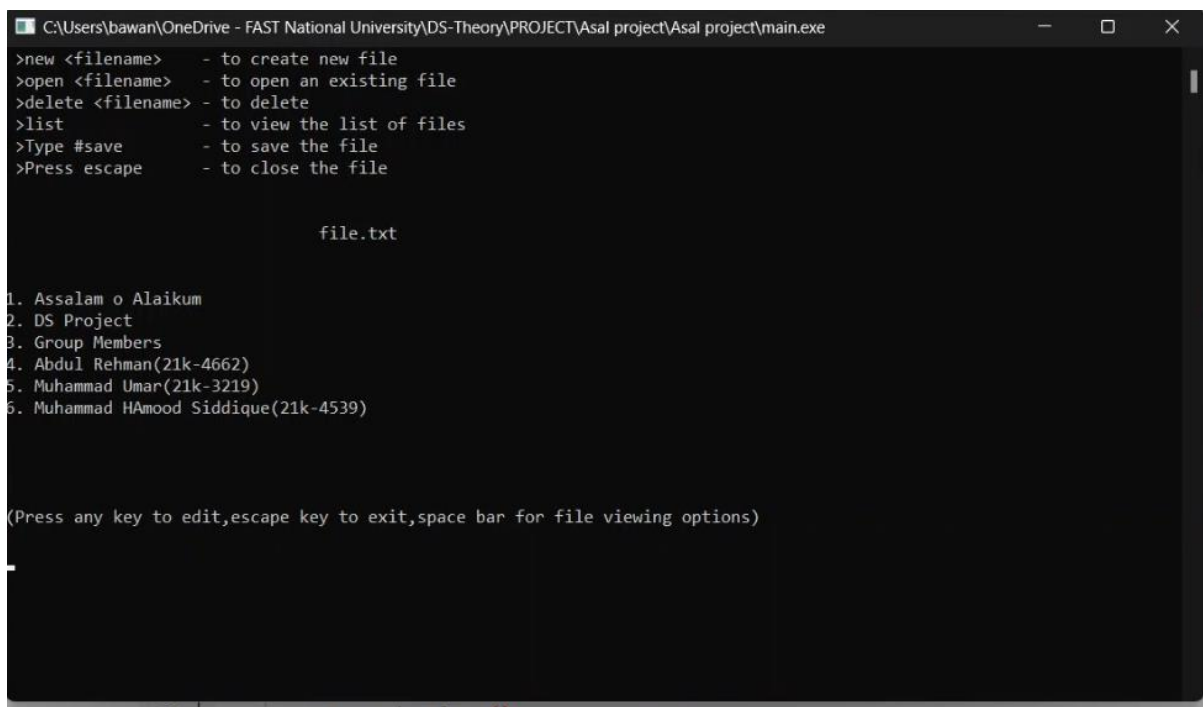
- Opening files:



```
C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe

>new <filename>      - to create new file
>open <filename>     - to open an existing file
>delete <filename>   - to delete
>list                - to view the list of files
>Type #save          - to save the file
>Press escape        - to close the file

new file
$:open file_
```



```
C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe

>new <filename>      - to create new file
>open <filename>     - to open an existing file
>delete <filename>   - to delete
>list                - to view the list of files
>Type #save          - to save the file
>Press escape        - to close the file

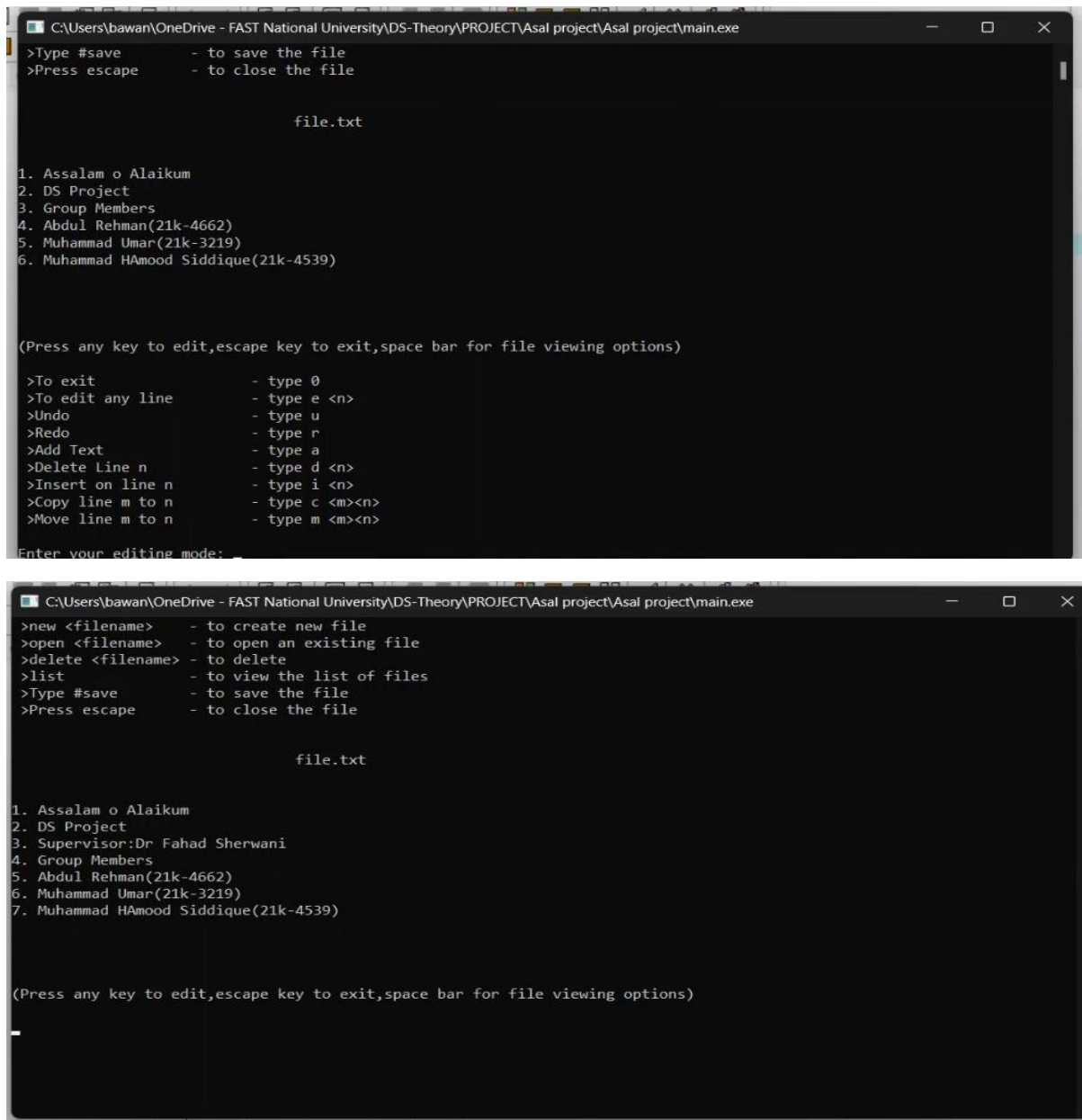
file.txt

1. Assalam o Alaikum
2. DS Project
3. Group Members
4. Abdul Rehman(21k-4662)
5. Muhammad Umar(21k-3219)
6. Muhammad HAmod Siddique(21k-4539)

(Press any key to edit,escape key to exit,space bar for file viewing options)
```

user can open an already existing text file with the "open" keyword and specifying the file name. The program will open and display the file if the file exists else will give an error message box "no such file exists."

- Editing:



The first screenshot shows a command-line window titled "C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe". It displays a menu of editing options: >Type #save - to save the file, >Press escape - to close the file, file.txt, a list of 6 items, (Press any key to edit,escape key to exit,space bar for file viewing options), and a detailed list of editing commands: >To exit - type 0, >To edit any line - type e <n>, >Undo - type u, >Redo - type r, >Add Text - type a, >Delete Line n - type d <n>, >Insert on line n - type i <n>, >Copy line m to n - type c <m><n>, >Move line m to n - type m <m><n>. The prompt "Enter your editing mode:" is at the bottom.

The second screenshot shows the same window after opening a file. The menu is: >new <filename> - to create new file, >open <filename> - to open an existing file, >delete <filename> - to delete, >list - to view the list of files, >Type #save - to save the file, >Press escape - to close the file. The file content is displayed: file.txt, a list of 7 items (the 6th item is now "Muhammad Umar(21k-3219)"), (Press any key to edit,escape key to exit,space bar for file viewing options), and a cursor at the bottom.

the user can enter the editing menu by pressing any key other than space and esc after successfully opening a file. the file contents will be displayed throughout the edit functions so the user can easily notice changes made. User can simply interact with the program by entering correctly the simple commands displayed on the menu screen. The edit line and delete line functions take a single parameter for the user defined line number whereas copy and move functions take two parameters for user defined input source and output destination. The undo and redo functions don't take any argument and simply keep operating upon the last performed step. The user can also add text either at the end of the text file or at any specified position.

- Viewing:

```

C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe
>open <filename> - to open an existing file
>delete <filename> - to delete
>list - to view the list of files
>type #save - to save the file
>Press escape - to close the file

file.txt

1. Assalam o Alaikum
2. Data Structures Course Project
3. Supervisor:Dr Fahad Sherwani
4. Group Members
5. Abdul Rehman(21k-4662)
6. Muhammad Umar(21k-3219)
7. Muhammad HAmood Siddique(21k-4539)
8. Thank you

(Press any key to edit,escape key to exit,space bar for file viewing options)

>To exit - type 0
>complete file - type p <0>
>line n - type p <n>
>line m to n - type t <m><n>

Enter your viewing mode:
24 commands.close();
25 cout<<"\n $:";

```

```

C:\Users\bawan\OneDrive - FAST National University\DS-Theory\PROJECT\Asal project\Asal project\main.exe

file.txt

4.Group Members
5.Abdul Rehman(21k-4662)
6.Muhammad Umar(21k-3219)
7.Muhammad HAmood Siddique(21k-4539)

```

the user can enter the viewing functions window by pressing the space bar key and will get options to print any specific line or multiple lines at a time with specified position or displaying the complete file by entering the commands shown on the console.



## **CONCLUSION:**

The text-editor project in C++ implements multiple data structures to store and process data providing ease of access to the user. The project allows the user to interact with the program through commands rather than using mouse or moving cursor here and there you could just simply get you work done by entering a simple command. The project has been kept simple and unnecessary design elements have been avoided to keep the code and the program short and precise and let the user fully enjoy the functionality of the editor. The use of linked list data structure makes it easy and quick to change text at any specified position and the LIFO principle using Stack makes it real easy to implement undo and redo functions. Although this is no way a perfect project, but the project still demonstrates a good knowledge of Data Structures. The code is easily readable and understandable and with the custom implementation of data structures, it is really easy for a fellow programmer to add more functionality into the project as per his/her wish list.