TEXT EDITOR

A PROJECT REPORT

SUBMITTED IN COMPLETE FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

[INFORMATION TECHNOLOGY]

Submitted by:

ARYAN MITTAL

2K20/IT/33

DHRUV KHERA

2K20/IT/43

Under the supervision of

Dr. Ritu Agarwal

INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(FORMERLY Delhi College of Engineering)

Bawana Road, Delhi-110042

NOVEMBER 2021

DELHI TECHNOLOGICAL UNIVERSITY

(FORMERLY Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We, (Aryan Mittal, Dhruv Khera , 2K20/IT/33, 2K20/IT/43) students of B. Tech. (Information Technology) hereby declare that the project Dissertation titled "Text Editor" which is submitted by us to the Department of Computer Science & Engg. , Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date:

Aryan Mittal(2K20/IT/33)

Dhruv Khera(2K20/IT/43)

INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(FORMERLY Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project Dissertation titled ”Text Editor“ which is submitted by ARYAN MITTAL, DHRUV KHERA (2K20/IT/33, 2K20/IT/43) [Information technology], Delhi Technological University, Delhi in complete fulfilment of the requirement for the award of the degree of the Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place:Delhi **Dr. Ritu aggarwal**

Date: **07/11/21**

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

In performing our major project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude toDr. Ritu Aggarwal, Mentor for the major project. Giving us a good guideline for report throughout numerous consultations. We would also like to extend our deepest gratitude to all those who have directly and indirectly guided us in writing this project.

Many people, our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us inspiration to improve our assignment.

We thank all the people for their help directly and indirectly to complete our assignment. In addition, we would like to thank the Department of Computer Science & Engg. Delhi Technological University for giving us the opportunity to work on this topic.

INTRODUCTION

We all have used a text editor from the beginning of our education. A text editor is simply a type of computer program that edits plain text. So, in this project we will be making the text editor which is a must needed program in every device, say laptops, desktops, mobiles, PC's. The project implements a text editor using Linked List and Stack data structures.
Each line of text is saved into a linked list node, which consists of a data portion holding the text and a pointer to the next node of the linked list, which is the next line of the text. The program makes use of Stack data structure to implement the Undo function.

Features:

- Open an existing file and parse every line to a linked list.
- Save into a file
- Insert text into line n (user given line number and text)
- Delete line n (user given line number)
- Move line n to line m (interchanging two user given lines)
- Replace text in line n (user given line number and text) o Next page
- Previous page
- Undo (Implemented using Stack data structure)

We all have used various text editors on our day to day work.

For eg:

1. Notepad++ is a text and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window.
2. The Xcode IDE is at the center of the Apple development experience. ... Sublime Text and Xcode are primarily **classified as "Text Editor"** and "Integrated Development Environment" tools respectively.

Uses of Header Files

fstream

This file has been used to perform file handling operations.

iostream

This file is used for using basic input and output operations like cin,cout.

stack

For creating a stack, we include the <stack> header file in our code.

string

This file is used for functions related to string datatype.

CODING IMPLEMENTATION

We already had prerequisite knowledge of programming in C++ , with topics like basics of C++ , file handling, Object oriented programming, switch case, loops, Data Structures like stacks, Linked Lists, etc. We used the gcc compiler to execute the code.

We created our program by implementing classes to make objects which are further used to write the data in the Linked List named “linked_list” .

FLOW OF PROGRAM:

Insert text into line n (user given line number and text)

A linked list is used to implement the insert text into line N feature of the text editor. In this feature, the user is first asked the line number in which he wants to insert text and then the text he wants to be inserted.

If the line number entered already exists, the text is added into the previously saved text. Or else a new line is made and text is added. Special functions are created to implement all these options.

Created a counter to keep track of the number of lines till now.

Code:

```
if (choice == 1)            //Insertion of a line, any line, works fine
{
    int lineGiven;
    string dataGiven;
    cout<<"Enter line you want the text to be placed into : ";
    cin >> lineGiven;
    cout<<"Enter text : ";
    cin.ignore(1);
    getline(cin,dataGiven);
    dataGiven+="\n";
    if (lineGiven == 1)
    {
        addToHead(dataGiven);
    }
    else if (lineGiven > numOfLines)
    {
        insertFurtherAway(dataGiven,lineGiven);
    }
    else if (lineGiven < numOfLines)
    {
        insertTextInBetween(dataGiven,lineGiven);
    }
    else if (lineGiven == numOfLines)
    {
        int selection;
        cout<<"Enter 1 to replace the last line, enter 2 to insert a new line";
        cin>>selection;
        if (selection == 1)
        {
            replaceTextInLine(dataGiven,lineGiven);
        }
        else if (selection == 2)
        {
            addToTail(dataGiven);
        }
    }
}
```

Functions used:

```
void addToHead(string dataGiven){       //this function will add to Head
    if (head == NULL)                   //no node, empty linked list
    {
        node *temp;
        temp = new node;
        temp->data = dataGiven;
        temp->next = NULL;
        head = temp;
        tail = head;
        numOfLines++;
    }
    else                                //one or more than one node
    {
        node *temp;
        temp = new node;
        temp->data = dataGiven;
        temp->next = NULL;
        temp->next = head;
        head = temp;
        numOfLines++;
    }
    undoCmd adddedToHead;
    adddedToHead.lineNumber = 1;
    adddedToHead.commandNumber = 1;
    undoStack.push(adddedToHead);
}
```

```
void insertFurtherAway(string dataGiven, int lineGiven)
    undoCmd insertedFurtherAway;
    insertedFurtherAway.lineNumber = 0;
    insertedFurtherAway.commandNumber = 9;
    if (head == NULL)
    {
        while(numOfLines < lineGiven-1)
        {
            whateverAddToTail("\n");
            insertedFurtherAway.lineNumber++;
        }
        // insertedFurtherAway.lineNumber++;
        whateverAddToTail(dataGiven);
    }
    else{
        while(numOfLines < lineGiven-1)
        {
            whateverAddToTail("\n");
            insertedFurtherAway.lineNumber++;
        }
        whateverAddToTail(dataGiven);
    }
    undoStack.push(insertedFurtherAway);
```

```
void replaceTextInLine(string dataGiven,int lineGiven){
    undoCmd replacedLine;
    if (numOfLines < lineGiven)
    {
        cout<<"The line you entered exceeds the existing number of lines..."<<endl;
    }
    else if (lineGiven == 0)
    {
        cout<<"There's no line 0, did you mean 1 ?"<<endl;
    }
    else if (numOfLines >= lineGiven )
    {
        node *temp = head;
        int goToLine = 1;
        while(goToLine < lineGiven)
        {
            temp = temp->next;
            goToLine++;
        }
        string backup = temp->data;
        temp->data = dataGiven;
        replacedLine.lineNumber = lineGiven;
        replacedLine.text = backup;
        replacedLine.commandNumber = 4;
        undoStack.push(replacedLine);
    }
}
```

```
void insertTextInBetween(string dataGiven, int lineGiven){
    if (lineGiven == 0)
    {
        cout<<"There's no line 0, did you mean 1?"<<endl;
    }
    else{
        node *prevNode = head;
        node *nextNode = head;
        node *temp = new node();
        temp->data = dataGiven;
        temp->next = NULL;
        int iterator = 2;
        while(iterator < lineGiven)
        {
            prevNode = prevNode->next;
            nextNode = nextNode->next;
            iterator++;
        }
        nextNode = nextNode->next;
        prevNode->next = temp;
        temp->next = nextNode;
        numOfLines++;
        undoCmd insertedInBetween;
        insertedInBetween.lineNumber = lineGiven;
        insertedInBetween.commandNumber = 6;
        undoStack.push(insertedInBetween);
    }
}
```

```
void addToTail(string dataGiven){       //this function will add to Tail
    if (head == NULL)                   //no node, empty linked list
    {
        node *temp;
        temp = new node;
        temp->data = dataGiven;
        temp->next = NULL;
        head = temp;
        tail = head;
        numOfLines++;
    }
    else                                //one or more than one node
    {
        node *temp;
        temp = new node;
        temp->data = dataGiven;
        temp->next = NULL;
        tail->next = temp;
        tail = temp;
        numOfLines++;
    }
    undoCmd addedToTail;
    addedToTail.lineNumber = 1;
    addedToTail.commandNumber = 8;
    undoStack.push(addedToTail);
}
```

Delete line n (user given line number)

Next feature of our text editor is the deletion of lines. User is asked the line number he wants to delete and according to that the line is deleted from the file.

To implement this 5 test cases are encountered

1) When no line exists

2) 1st line to be deleted

3) When line given== numoflines(total lines)

4) When line given is greater than total lines entered (does not exist)

5) Line given in range.

```
else if (choice == 2)       //Deletion of a line, any line, works fine
{
    int lineGiven;
    cout<<"Enter the line you want to delete : ";
    cin>>lineGiven;
    deleteLine(lineGiven);
}
```

Functions Used:

```
void deleteLine(int lineGiven){
    if (head == NULL)
    {
        cout<<"There is no line to be deleted/removed."<<endl;
    }
    else if(head == tail){
        node *temp = head;
        delete(temp);
        head = NULL;
        tail = NULL;
        numOfLines--;
    }
    else if(lineGiven == 0){
        cout<<"There's no line 0, did you mean 1?"<<endl;
    }
    else if(lineGiven == 1){
        string backup = head->data;
        node *temp = head;
        node *nextNode = head->next;
        head = nextNode;
        delete(temp);
        numOfLines--;
        undoCmd headRemoved;
        headRemoved.text = backup;
        headRemoved.lineNumber = 1;
        headRemoved.commandNumber = 12;
        undoStack.push(headRemoved);
    }
    else if(lineGiven == numOfLines){
        node *temp = head;
        undoCmd deletedLine;
        deletedLine.commandNumber = 11;
        while (temp->next != NULL && temp->next->next != NULL)
        {
            temp = temp->next;
        }
        tail = temp;
        string backup = temp->next->data;
        delete temp->next;
        temp->next = NULL;
```

```
        temp->next = NULL;
        numOfLines--;
        deletedLine.text = backup;
        deletedLine.lineNumber = lineGiven;
        undoStack.push(deletedLine);

    }
    else if (lineGiven > numOfLines)
    {
        cout<<"Entered line is larger than existing lines..."<<endl;
    }
    else if (lineGiven < numOfLines)
    {
        undoCmd deletedLine;
        deletedLine.commandNumber = 10;
        node *prevNode = head;
        node *nextNode = head;
        node *temp = head;
        int iterator = 2;
        while(iterator < lineGiven)
        {
            prevNode = prevNode->next;
            nextNode = nextNode->next;
            iterator++;
        }
        nextNode = nextNode->next;
        temp = nextNode;
        nextNode = nextNode->next;
        prevNode->next = nextNode;
        string backup = temp->data;
        delete(temp);
        numOfLines--;
        deletedLine.text = backup;
        deletedLine.lineNumber = lineGiven;
        undoStack.push(deletedLine);
    }
}
```

Move line n to line m (interchanging two user given lines)

This feature swaps two lines of the code. User is asked to enter line numbers for the lines swapping wants to be done.

Code:

```
else if (choice == 3)       //Interchanging two lines, any two line, works fine
{
    int lineGiven1;
    int lineGiven2;
    cout<<"Enter line 1 you want to swap : ";
    cin>>lineGiven1;
    cout<<"Enter line 2 you want to swap : ";
    cin>>lineGiven2;
    moveNtoM(lineGiven1, lineGiven2);
}
```

Functions Used:

```
void moveNtoM(int nLineGiven, int mLineGiven){
    if (nLineGiven == 1)
    {
        string headText = head->data;
        deleteHead();
        insertTextInBetween(headText,mLineGiven);
    }
    else
    {
        node *temp = head;
        int iterator = 1;
        while(iterator < nLineGiven)
        {
            temp = temp -> next;
            iterator++;
        }
        string dataSaved = temp->data;
        deleteLine(nLineGiven);
        insertTextInBetween(dataSaved,mLineGiven);
    }
    undoCmd moveHeadToM;
    moveHeadToM.commandNumber = 2;
    moveHeadToM.nLine = nLineGiven;
    moveHeadToM.mLine= mLineGiven;
    undoStack.push(moveHeadToM);
}
```

Replace text in line n (user given line number and text)

We make mistakes, in a text editor, replacing features is a must. Here, the user gives the line number, and the already existing line context is deleted and the new text is added to it.

```
else if (choice == 4)
{
    int lineGiven;
    string dataGiven;
    cout<<"Enter line you want to change the content of : ";
    cin>>lineGiven;
    if (lineGiven > numOfLines)
    {
        cout<<"The line you entered exceeds the existing number of lines..."<<endl;
    }
    else{
        cout<<"Enter the new text : ";
        cin>>dataGiven;
        dataGiven+="\n";
        replaceTextInLine(dataGiven, lineGiven);
    }
}
```

Functions Used:

```
void replaceTextInLine(string dataGiven,int lineGiven){
    undoCmd replacedLine;
    if (numOfLines < lineGiven)
    {
        cout<<"The line you entered exceeds the existing number of lines..."<<endl;
    }
    else if (lineGiven == 0)
    {
        cout<<"There's no line 0, did you mean 1 ?"<<endl;
    }
    else if (numOfLines >= lineGiven )
    {
        node *temp = head;
        int goToLine = 1;
        while(goToLine < lineGiven)
        {
            temp = temp->next;
            goToLine++;
        }
        string backup = temp->data;
        temp->data = dataGiven;
        replacedLine.lineNumber = lineGiven;
        replacedLine.text = backup;
        replacedLine.commandNumber = 4;
        undoStack.push(replacedLine);
    }
}
```

Print all

This feature prints all the lines ,that is, the full linked list.

Code:

```
else if (choice == 5)       //Printing the whole list, works fine
{
    printall();
}
```

Functions Used:

```
void printall(){                        //function used to print the whole linked list
    node *temp = head;
    int linePrinted = 1;
    int pagePrinted = 2;
    int choice;
    if (head == NULL)
    {
        cout<<"no elements here, yay!"<<endl;
    }
    else{
        while(temp!=NULL)
        {
            if (linePrinted == 1)
            {
                cout<<"-------------------Page "<<"1"<<"-------------------\n";
            }
            else if ((linePrinted-1) % 10 == 0)
            {
                cout<<"-------------------Page "<<pagePrinted<<"-------------------\n";
                pagePrinted++;

            }
            cout<<linePrinted<<") "<<temp->data<<endl;
            temp = temp->next;
            linePrinted++;
        }
    }
}
```

Save into a .txt file

This feature saves your text into a .txt file at the place where the user's codes are saved. User is asked to enter the name of the file and then he/she can find the saved .txt file in their codes folder .

Code:

```
else if (choice == 6)       //Saving the list into a txt file, works fine
{
    saveAll();
}
```

Functions Used:

```
void saveAll(){
    node *temp = head;
    int linePrinted = 1;
    int pagePrinted = 2;
    string fileName;
    cout<<"Enter the file name : ";
    cin>>fileName;
    fileName+=".txt";
    outfile.open(fileName, ios_base::app);
    while(temp!=NULL)
    {
        outfile<<temp->data;
        temp = temp->next;
        linePrinted++;
    }
    outfile.flush();
    outfile.close();
}
```

Undo

We make mistakes, in a text editor, undo feature is a must. Here, the last operation the user performed gets reversed and you get back the previous linked list before executing the operation.

Code:

```
else if (choice == 7)
{
    if (undoStack.empty())
    {
        cout<<"No command."<<endl;

    }
    else{
        //cout<<"under construction..."<<endl;
        undo();

    }
}
```

Functions Used:

```
void undo(){                        //function used to undo the last action taken
    undoCmd temp = undoStack.top();
    if (temp.commandNumber == 1)
    {
        cout<<"Added To head, removing from head..."<<endl;
        deleteHead();
        undoStack.pop();
    }
    else if (temp.commandNumber == 2)
    {
        cout<<"Moved M to N, moving N to M"<<endl;
        moveNtoM(temp.mLine, temp.nLine);
        undoStack.pop();
    }
    else if (temp.commandNumber == 3)
    {
        cout<<"Deleted head, replacing head..."<<endl;;
        addToHead(temp.text);
        undoStack.pop();
    }
    else if (temp.commandNumber == 4)
    {
        cout<<"Replaced line, replacing again..."<<endl;
        replaceTextInLine(temp.text,temp.lineNumber);
        undoStack.pop();
    }
    else if (temp.commandNumber == 5)
    {
        cout<<"Inserted to Head, removing from head..."<<endl;
        deleteHead();
```

```
else if (temp.commandNumber == 5)
{
    cout<<"Inserted to Head, removing from head..."<<endl;
    deleteHead();
    undoStack.pop();
}
else if (temp.commandNumber == 6)
{
    cout<<"Inserted in between, removing that line..."<<endl;
    deleteLine(temp.lineNumber);
    undoStack.pop();
}
else if (temp.commandNumber == 7)
{
    cout<<"Deleted Tail, inserting again..."<<endl;
    addToTail(temp.text);
    undoStack.pop();
}
else if (temp.commandNumber == 8)
{
    cout<<"Added to tail, removing from tail..."<<endl;
    deleteTail();
    undoStack.pop();
}
else if (temp.commandNumber == 9)
{
    int whatever = temp.lineNumber;
    while(whatever >= 0){
        whateverDeleteTail();
        whatever--;
    }
```

```
    else if (temp.commandNumber == 9)
    {
        int whatever = temp.lineNumber;
        while(whatever >= 0){
            whateverDeleteTail();
            whatever--;
        }
        undoStack.pop();
    }
    else if (temp.commandNumber == 10)
    {
        cout<<"Line deleted, inserting again..."<<endl;
        insertTextInBetween(temp.text, temp.lineNumber);
        undoStack.pop();
    }
    else if (temp.commandNumber == 11)
    {
        cout<<"Last line deleted, inserting again..."<<endl;
        addToTail(temp.text);
        undoStack.pop();
    }
    else if (temp.commandNumber == 12)
    {
        cout<<"First line deleted, inserting again..."<<endl;
        addToHead(temp.text);
        undoStack.pop();
    }
}
```

Open a .txt file

This feature opens your a .txt file stored at the place where the user's codes are saved. User is asked to enter the name of the file and then the text in the file is saved in the form of nodes in our Linked List.

Code:

```
else if (choice == 8)
{
    node* current = head;
    node* next;
    while (current != NULL)
    {
        next = current->next;
        free(current);
        current = next;
    }
    head = NULL;
    openFile();
}
```

Functions Used:

```
void openFile(){
    string fileName;
    cout<<"Enter the file name : ";
    cin>>fileName;
    fileName+=".txt";
    ifstream myfile;
    myfile.open(fileName);
    string s;
    while(getline(myfile,s))
    {
        addToTail(s);
    }
    myfile.close();
}
```

Print the next Page

This feature prints all the lines on the next page from the current one .

Code:

```
else if (choice == 9)        //Printing the next page
{
    if (prevPagePrinted*10 > numOfLines)
    {
        // cout<<"No more page left to print."<<endl;
        printOnePage(prevPagePrinted);

    }
    else if (prevPagePrinted == 1)
    {
        printOnePage(1);
        prevPagePrinted++;

    }
    else{
        printOnePage(prevPagePrinted);
        prevPagePrinted++;

    }
}
```

Functions Used:

```
void printOnePage(int pageGiven){
    node *temp = head;
    if (numOfLines < pageGiven*10)
    {
        int iterator = 1;
        while(iterator < (pageGiven*10)-9){
            temp = temp->next;
            iterator++;
        }
        for (int start = (pageGiven*10)-9 ; start <= numOfLines; start++)
        {
            cout<<start<<") "<<temp->data<<endl;
            temp = temp->next;
        }
        cout<<"------------------Page "<<pageGiven<<"------------------\n";
    }
    else if (numOfLines >= pageGiven * 10)
    {
        int iterator = 1;
        while(iterator < (pageGiven*10)-9){
            temp = temp->next;
            iterator++;
        }
        for (int start = (pageGiven*10)-9 ; start <= pageGiven*10; start++)
        {
            cout<<start<<") "<<temp->data<<endl;
            temp = temp->next;
        }
        cout<<"------------------Page "<<pageGiven<<"------------------\n";
    }
    else if (pageGiven * 10 > numOfLines)
    {
        cout<<"WHOOSH, you want to print an inexisting page, collect yourself!"<<endl;
    }
}
```

Print the previous Page

This feature prints all the lines on the previous page from the current one .

Code:

```
else if (choice == 10)          //Printing the previous page
{
    if (prevPagePrinted <= 0)
    {
        prevPagePrinted = 1;
        printOnePage(1);

    }
    else if (prevPagePrinted == 1)
    {
        prevPagePrinted--;
        printOnePage(1);

    }
    else{
        prevPagePrinted--;
        printOnePage(prevPagePrinted);

    }
}
```

Functions Used:

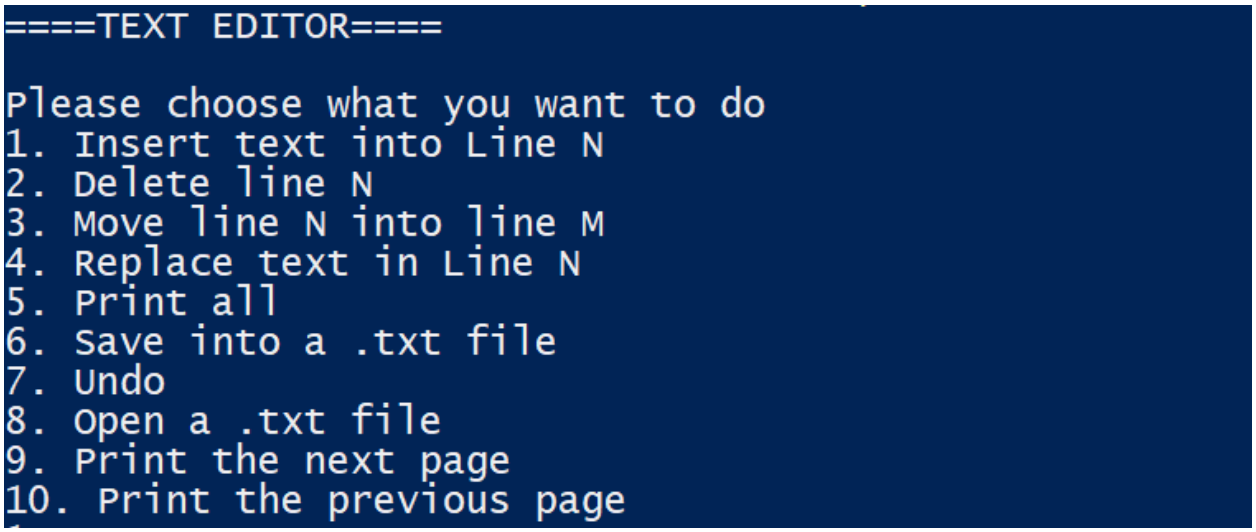
```
void printOnePage(int pageGiven){
    node *temp = head;
    if (numOfLines < pageGiven*10)
    {
        int iterator = 1;
        while(iterator < (pageGiven*10)-9){
            temp = temp->next;
            iterator++;
        }
        for (int start = (pageGiven*10)-9 ; start <= numOfLines; start++)
        {
            cout<<start<<") "<<temp->data<<endl;
            temp = temp->next;
        }
        cout<<"------------------Page "<<pageGiven<<"------------------\n";
    }
    else if (numOfLines >= pageGiven * 10)
    {
        int iterator = 1;
        while(iterator < (pageGiven*10)-9){
            temp = temp->next;
            iterator++;
        }
        for (int start = (pageGiven*10)-9 ; start <= pageGiven*10; start++)
        {
            cout<<start<<") "<<temp->data<<endl;
            temp = temp->next;
        }
        cout<<"------------------Page "<<pageGiven<<"------------------\n";
    }
    else if (pageGiven * 10 > numOfLines)
    {
        cout<<"WHOOSH, you want to print an inexisting page, collect yourself!"<<endl;
    }
}
```

RESULT / OUTPUTS

Here is our intro page !

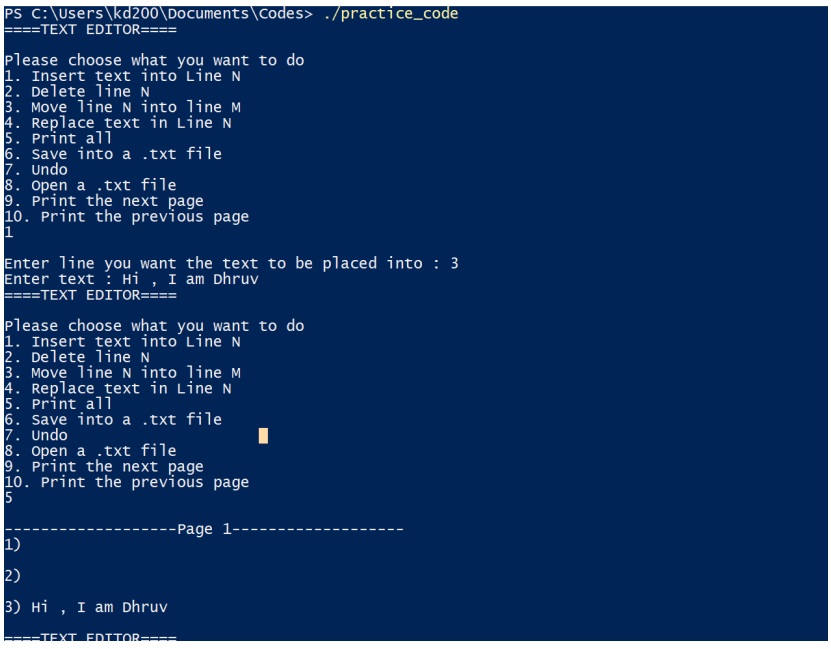
In this screen, we are showing the user all the tasks which he can

Perform using the program :



Screenshots of various choices :

Choice 1:



Choice 2:

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 3
Enter text : Aryan and dhruv
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
```

```
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
2

Enter the line you want to delete : 3
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5
```

```
------------------Page 1------------------
1)

2)

====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
0



...Program finished with exit code 0
Press ENTER to exit console.
```

Choice 3:

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 3
Enter text : Aryan is killer
```

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

------------------Page 1------------------
1)

2)

3) Aryan is killer
```

```
Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
3

Enter line 1 you want to swap : 3
Enter line 2 you want to swap : 1
====TEXT EDITOR====
```

```
Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

------------------Page 1------------------
1) Aryan is killer

2)

3)
```

Choice 4:

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 3
Enter text : i am cristiano
====TEXT EDITOR====
```

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
4

Enter line you want to change the content of : 3
Enter the new text : i am messi
====TEXT EDITOR====
```

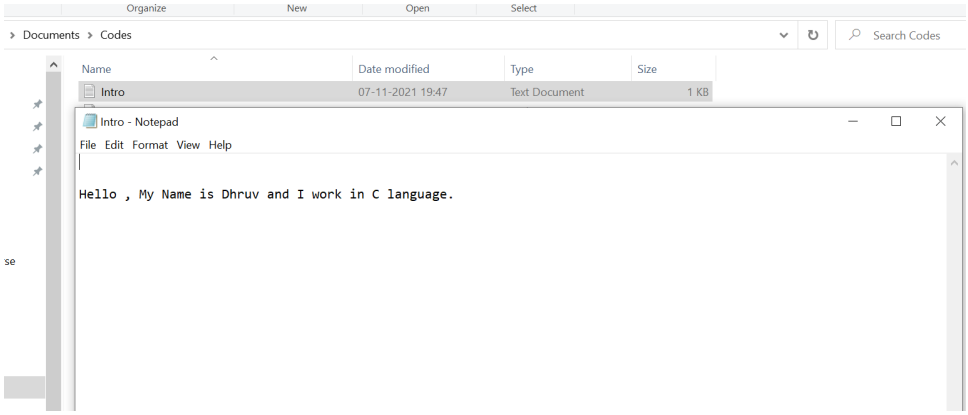
```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page



...Program finished with exit code 0
Press ENTER to exit console.
```

Choice 5:

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 4
Enter text : Lets Print the whole List
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

--------------------Page 1--------------------
1)

2)

3)

4) Lets Print the whole List
```

Choice 6:

```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 3
Enter text : Hello , My Name is Dhruv and I work in C language.
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
6

Enter the file name : Intro
====TEXT EDITOR====
```



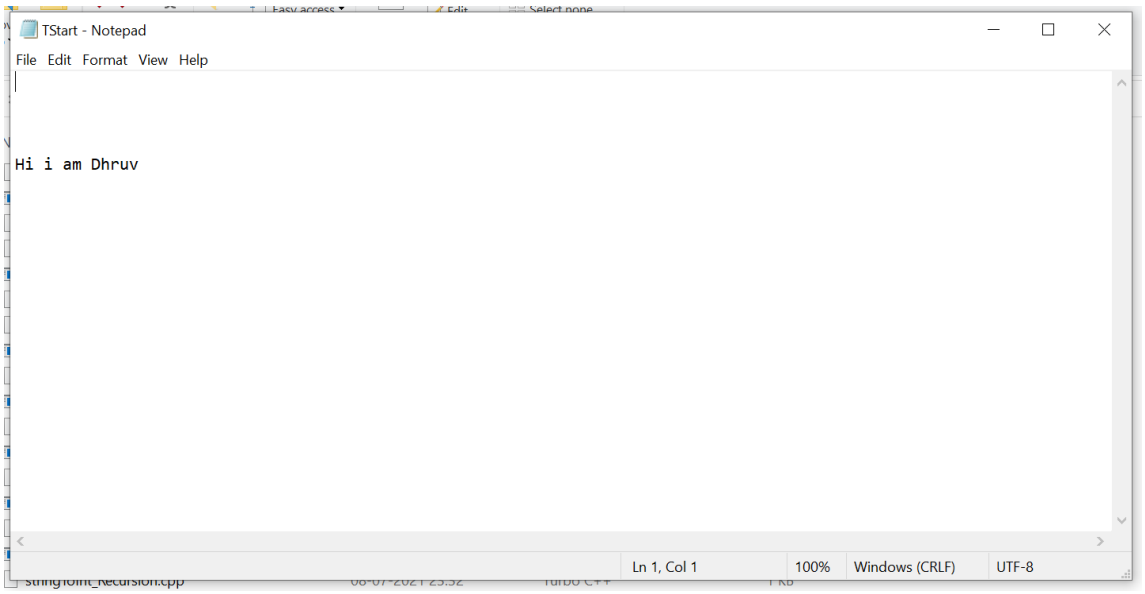
Choice 7:

```
--------------------Page 1--------------------
1)
2)
3)
4)
5) Hi i am Dhruv
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
1

Enter line you want the text to be placed into : 2
Enter text : I want to be a Developer !
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5
```

```
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

--------------------Page 1--------------------
1)
2) I want to be a Developer !

3)
4)
5)
6) Hi i am Dhruv
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
7

Inserted in between, removing that line...
====TEXT EDITOR====
```

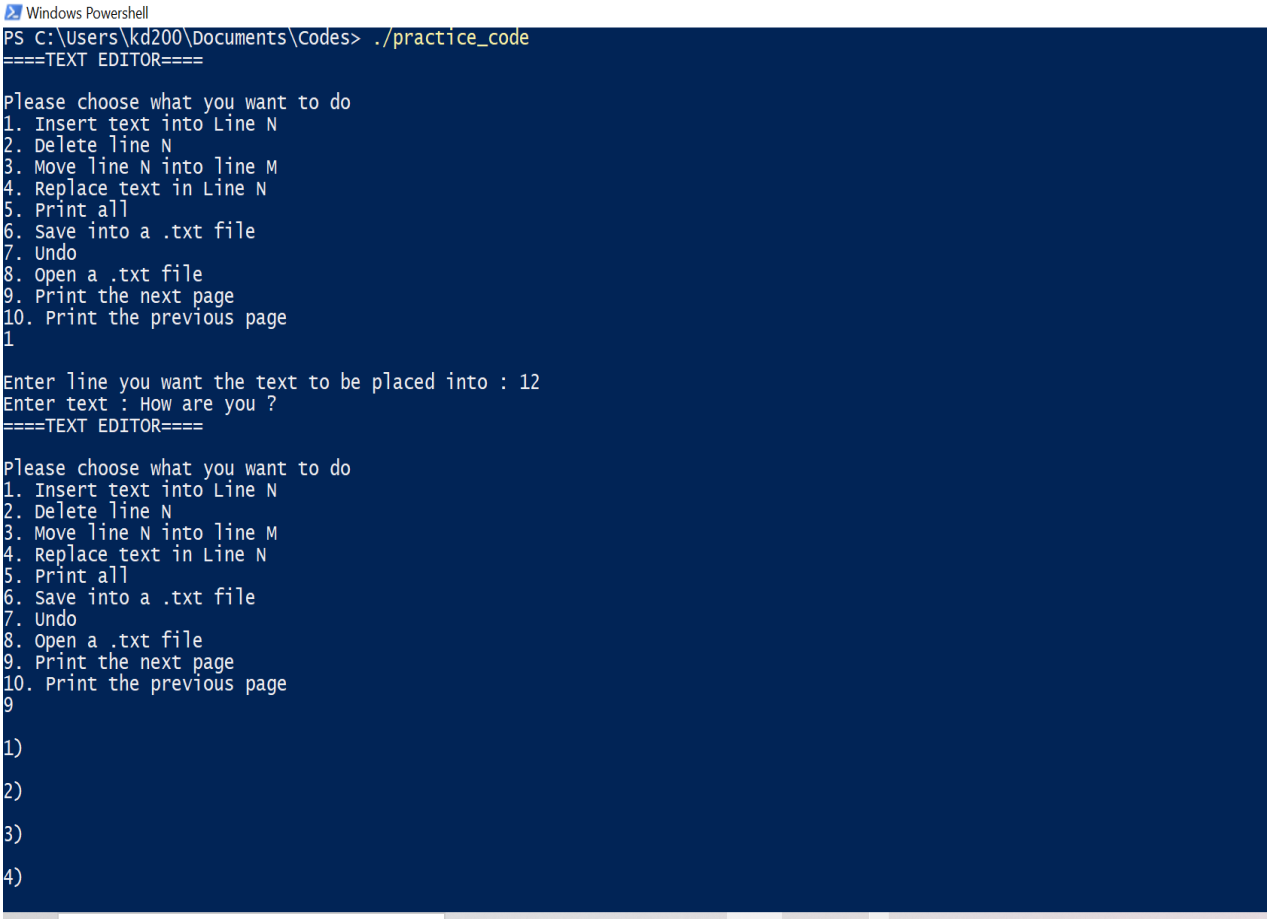
```
9. Print the next page
10. Print the previous page
7

Inserted in between, removing that line...
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

--------------------Page 1--------------------
1)
2)
3)
4)
5) Hi i am Dhruv
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
```

Choice 8:



```
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
8

Enter the file name : TStart
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

--------------------Page 1--------------------
1)
2)
3)
4)
5) Hi i am Dhruv
====TEXT EDITOR====
```

Choice 9:



```
Windows Powershell

2)

3)

4)

5)

6)

7)

8)

9)

10)

--------------------Page 1--------------------
====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
9

11)

12) How are you ?

--------------------Page 2--------------------
====TEXT EDITOR====
```

Choice 10:

```
Windows Powershell
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
5

--------------------Page 1--------------------
1)

2)

3) Hi , I am Dhruv

====TEXT EDITOR====

Please choose what you want to do
1. Insert text into Line N
2. Delete line N
3. Move line N into line M
4. Replace text in Line N
5. Print all
6. Save into a .txt file
7. Undo
8. Open a .txt file
9. Print the next page
10. Print the previous page
10

1)

2)

3) Hi , I am Dhruv

--------------------Page 1--------------------
```

SUMMARY

The primary aim of this “Text Editor” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a field like programming. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software. Anybody who wants to store some data or write code for programs can use a Text Editor to fulfil his needs. He has to be familiar with no technologies to use the Text Editor as it's very simple to use.

This Text Editor also allows users to open an existing file and parse every line to a linked list, Save into a file, Insert text into line n, Delete line n, Move line n to line m, Replace text in line n, Print next page ,Print previous page, Undo .

CONCLUSION

In this way we were able to accomplish our project (Bank Account Management). Throughout our work , we became comfortable with C++ programming using Data Structures and Algorithms.

We have learned so many things by working on this project on Text Editor like using classes , files , Stacks , Linked Lists, etc.and gained their practical Knowledge.

We gained basic knowledge about decision control statements and calling functions according to their scope . Also we polished our concept of constructors Time Complexity of stacks , Linked Lists and many more. We also got to know how to use different writing practices in c++ so that our code looks nice and clean with user friendliness and easily understandable .

FUTURE WORK

The “Text Editor” is a big and ambitious project. I am thankful for being provided this great opportunity to work on it. As already mentioned, this project has gone through extensive hard work. On the basis of the work, we have successfully designed and implemented a Text Editor. To know what the future of online editors looks like, it’s probably worth looking at the present – text editors aren't new. When you think of text editors, you probably think about an application (either on a desktop, laptop or mobile), and then an interface that lets you write something with your keyboard and view your various text files and style them according to your needs. And you’re not wrong either. The most valuable future looks are following below:

- ➢ More types of the Text Editors, maybe it will be open for all or paid only, that means more text editors outside.
- ➢ Customer issues development based on their needs, so the help desk will be aware of their needs and easy to use.
- ➢ Developing a mobile App for text editors that helps users to do the obtained operations with security codes so that you can use the personal information and important details like passwords,etc.

REFERENCES

- PROGRAMING WITH C++ BY PRITI ARORA (BOOK)
- C++ Primer

  Book by Barbara E. Moo, Josée Lajoie, and Stanley B. Lippman(BOOK)
- https://www.oreilly.com/library/view/data-structure-using/9788131755662/xhtml/chapter006.xhtml
- https://www.cs.odu.edu/~zeil/cs361/sum18/Public/linkedlists/index.html
- https://www.geeksforgeeks.org/stack-in-cpp-stl/
- https://www.cplusplus.com/reference/stack/stack/
- https://data-flair.training/blogs/file-handling-in-c/