

## Project No.1: Simple Line Editor

Name	Student No.	Email	Responsibility
李志容	14346009	978282388@qq.com	Parts of codes, parts of the reports
谭笑	14346022	296309711@qq.com	Parts of codes, parts of the reports

Data: 2016.3.20

### 1. Introduction

This project is to write a simple line editor. The editor contains functions of opening a file, inserting text lines, deleting text lines, changing the current lines, printing the lines and saving the texts to a file. To solve these problems, we design a new class -- Editor. In the class, we complete each functions step-by-step using the data structure of list. Finally, through many tests, our project successfully completes all the functions we have set.

### 2. Analysis and Design

#### 1) Data structure: List

A list is a doubly linked list. It supports both forward and backward traversal, and constant time insertion and removal of elements at the beginning or the end, or in the middle. A vector is a sequence that supports random access to elements, constant time insertion and removal of elements at the end, but linear time insertion and removal of elements at the beginning or in the middle. As for stack and queue, they are not suitable here because it is hard to delete and insert elements in the middle of texts. In our project, insertion and removal are our main functions, so the list is the best choice since it has the minimal time complexity. Finally, we decided to use list.

#### 2) Important algorithms

Since most parts of our program use the functions of List to solve problems, the algorithms are not so much and complex. There is an algorithm which we use to get the parameters in some commands, such as \$Delete k m, \$Line k, \$Print k m. The implement is:

```
bool Editor::str_to_int(string s, int a[], int n) {
    int len = s.length();           //the length of string
    int i = 0, j = 0, count = 0;    //count the numbers of integers extracted
    for (; j < n && i < len; j++) {
        count++;                    //while encounters a blank, count++
        a[j] = 0;
        while (s[i] != ' ' && i < len) { //while encounters a blank, while loop stops
            if (!isdigit(s[i])) return false;
            a[j] = a[j] * 10 + (s[i] - '0'); //sum up to calculate the integer.
        }
    }
}
```

```

        i++;
    }
    i++;
}
if (count != n) return false;
return true;
}

```

string s is the string containing parameters of integers like "k m". int a[] is used to store the integers we get. n is the numbers of integers we need, checking whether the numbers of integers we extract is correct. If they are not equal, some information will be returned.

**Justification:** this algorithm is more efficient because it simultaneously converts numbers of string type to numbers of integer type and checks whether the numbers of parameters is correct. For example, if the command is \$Delete 3, it will return false since count is 1 and n is 2. Obviously, the time complexity is Linear.

### 3) Time complexity

Firstly we analyze each function's complexity. For each command, the insertion function cause constant time based on STL function of List. The deletion and print function cause linear time  $O(m - k)$ , which the maximum time will cause  $O(\text{text.size}())$ . The open and save function also cause linear time  $O(\text{text.size}())$  because saving texts to file or extracting from file to texts, it operates line-by-line. All above, the whole algorithms will cause a linear time complexity.

## 3. Test

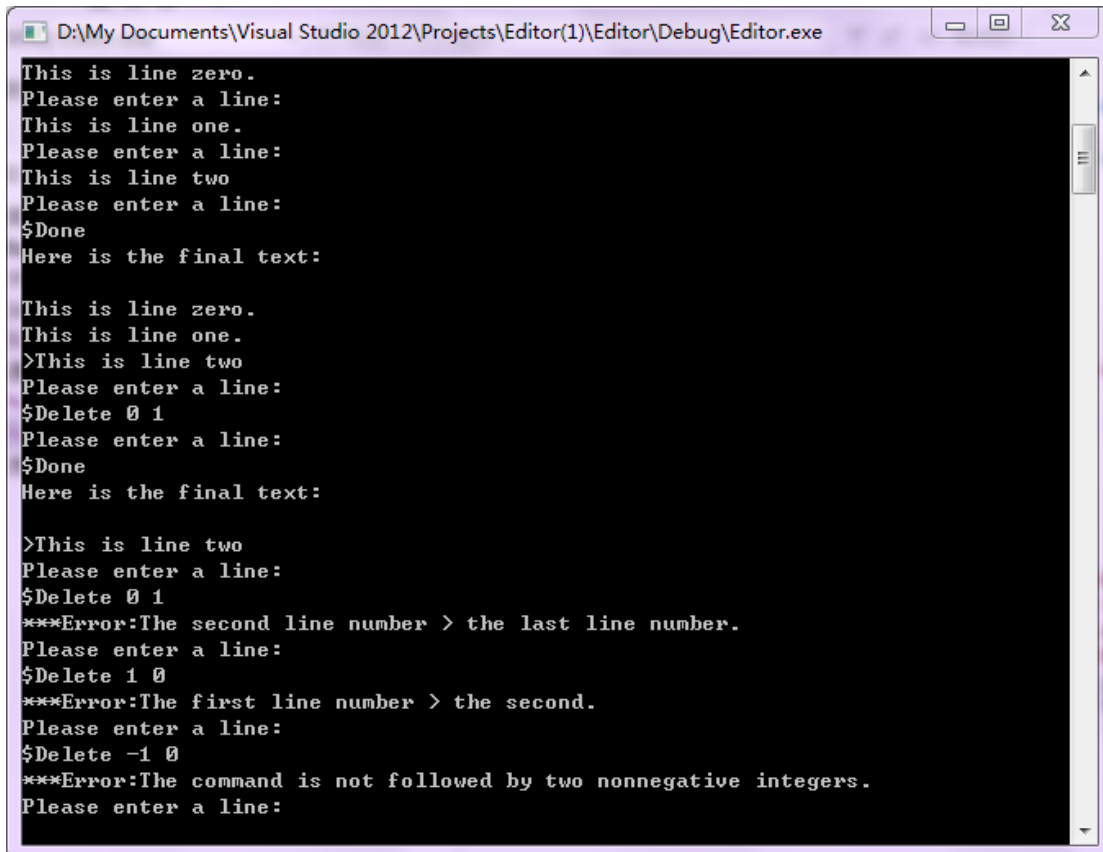
- 1) **Test \$Insert:** Input several lines. \$Done and show all the lines.

```

D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
Please enter a line:
$Insert
Please enter a line:
This is line zero.
Please enter a line:
This is line one.
Please enter a line:
This is line two
Please enter a line:
$Done
Here is the final text:
This is line zero.
This is line one.
>This is line two
Please enter a line:

```

- 2) **Test \$Delete:** Delete some lines, including errors. \$Done and show all the lines.

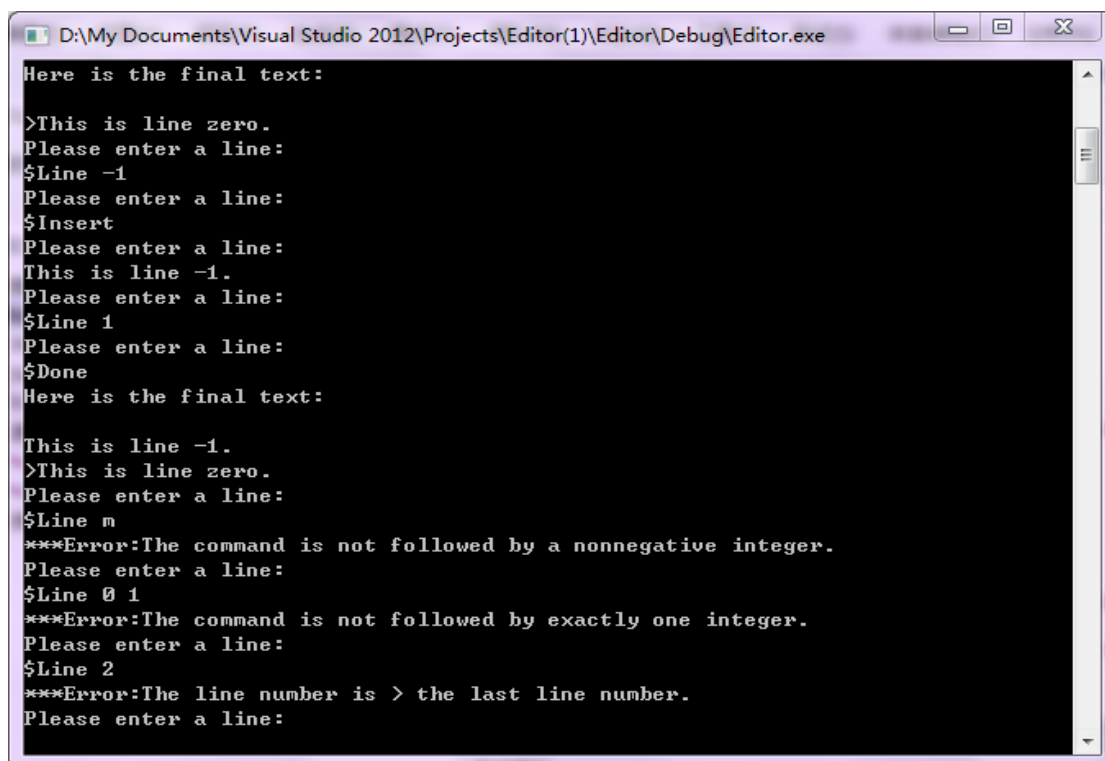


```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
This is line zero.
Please enter a line:
This is line one.
Please enter a line:
This is line two
Please enter a line:
$Done
Here is the final text:

This is line zero.
This is line one.
>This is line two
Please enter a line:
$Delete 0 1
Please enter a line:
$Done
Here is the final text:

>This is line two
Please enter a line:
$Delete 0 1
***Error:The second line number > the last line number.
Please enter a line:
$Delete 1 0
***Error:The first line number > the second.
Please enter a line:
$Delete -1 0
***Error:The command is not followed by two nonnegative integers.
Please enter a line:
```

- 3) **Test \$Line:** Change the current line, including errors, including case of  $m = -1$ .  
\$Done and show all the lines.

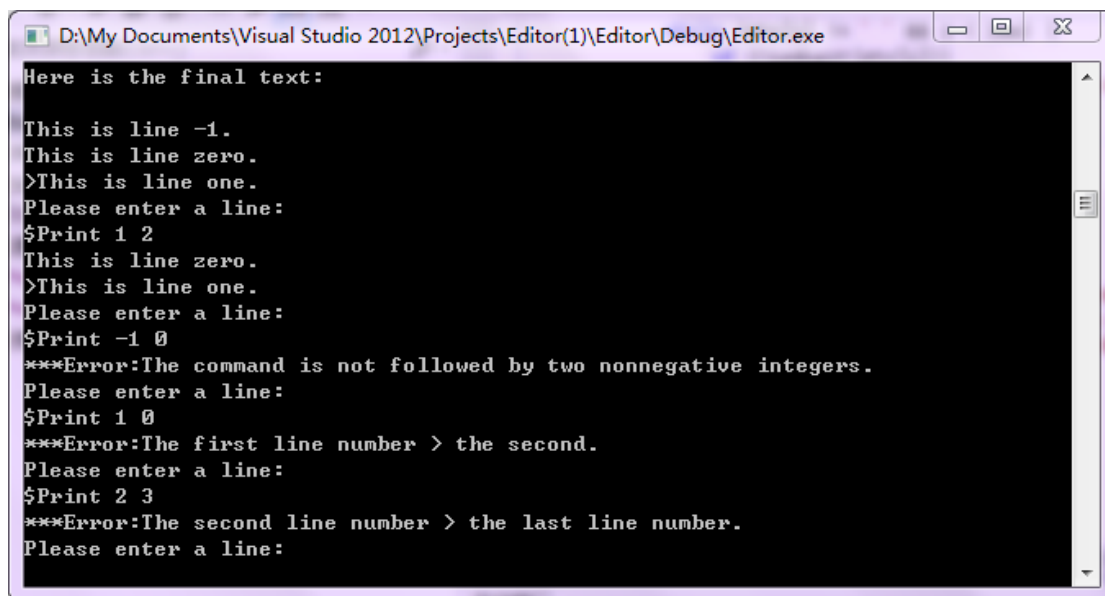


```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
Here is the final text:

>This is line zero.
Please enter a line:
$Line -1
Please enter a line:
$Insert
Please enter a line:
This is line -1.
Please enter a line:
$Line 1
Please enter a line:
$Done
Here is the final text:

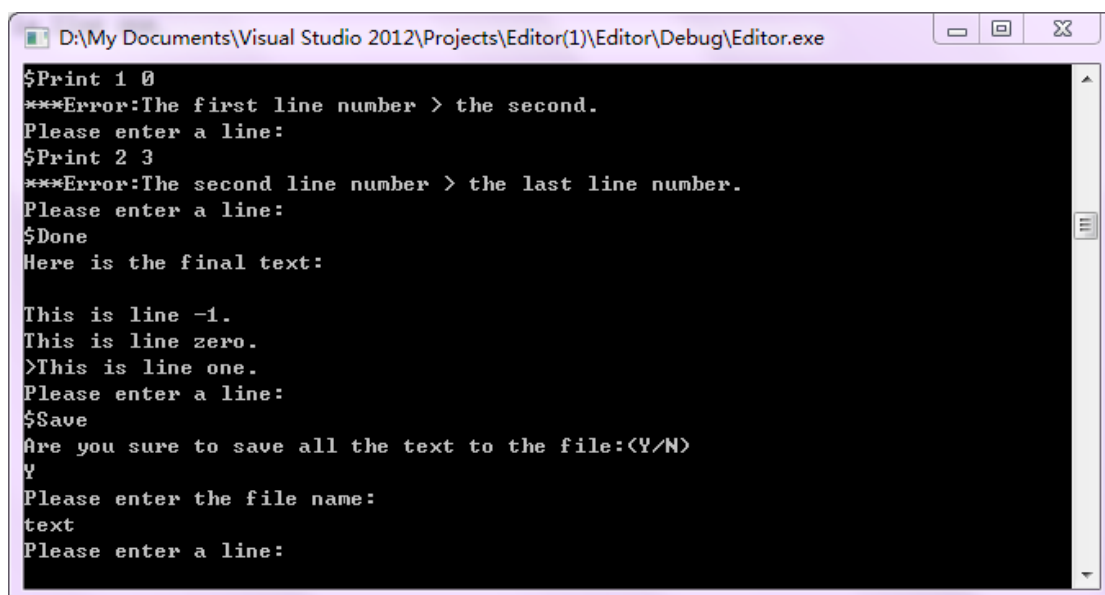
This is line -1.
>This is line zero.
Please enter a line:
$Line m
***Error:The command is not followed by a nonnegative integer.
Please enter a line:
$Line 0 1
***Error:The command is not followed by exactly one integer.
Please enter a line:
$Line 2
***Error:The line number is > the last line number.
Please enter a line:
```

4) **Test \$Print:** Print some lines, including errors.



```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
Here is the final text:
This is line -1.
This is line zero.
>This is line one.
Please enter a line:
$Print 1 2
This is line zero.
>This is line one.
Please enter a line:
$Print -1 0
***Error:The command is not followed by two nonnegative integers.
Please enter a line:
$Print 1 0
***Error:The first line number > the second.
Please enter a line:
$Print 2 3
***Error:The second line number > the last line number.
Please enter a line:
```

5) **Test \$Save:** Save the text in a file.



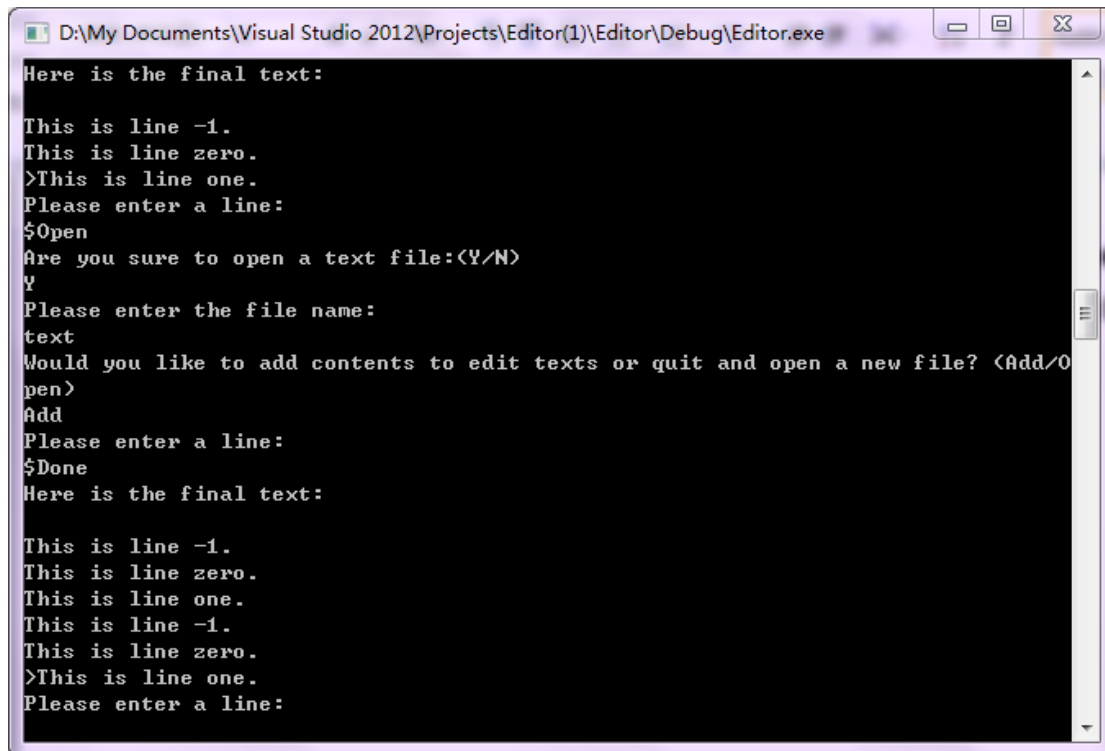
```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
$Print 1 0
***Error:The first line number > the second.
Please enter a line:
$Print 2 3
***Error:The second line number > the last line number.
Please enter a line:
$Done
Here is the final text:
This is line -1.
This is line zero.
>This is line one.
Please enter a line:
$Save
Are you sure to save all the text to the file:<Y/N>
Y
Please enter the file name:
text
Please enter a line:
```

Here is the saved file:



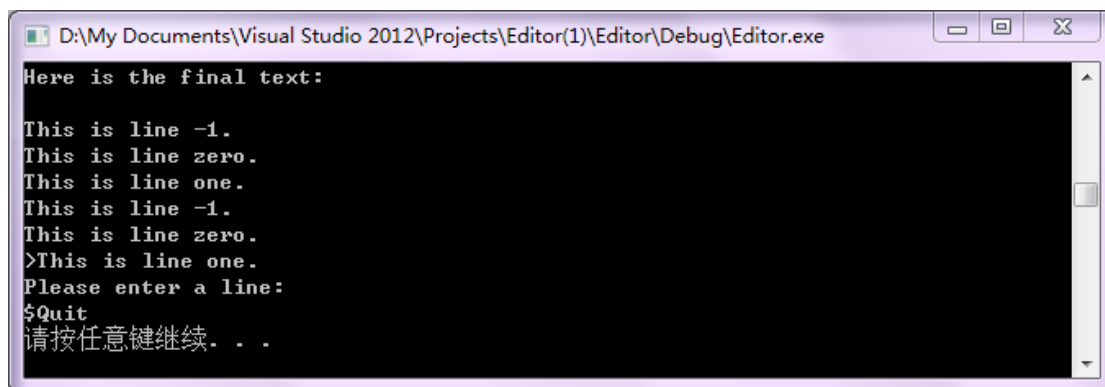
```
text - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
This is line -1.
This is line zero.
This is line one.
```

- 6) **Test \$Open:** Open the file we just saved and insert it into the text. \$Done and show all the lines.



```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
Here is the final text:
This is line -1.
This is line zero.
>This is line one.
Please enter a line:
$Open
Are you sure to open a text file:<Y/N>
Y
Please enter the file name:
text
Would you like to add contents to edit texts or quit and open a new file? <Add/Open>
Add
Please enter a line:
$Done
Here is the final text:
This is line -1.
This is line zero.
This is line one.
This is line -1.
This is line zero.
>This is line one.
Please enter a line:
```

- 7) **Test \$Quit:** Exit the program.



```
D:\My Documents\Visual Studio 2012\Projects\Editor(1)\Editor\Debug\Editor.exe
Here is the final text:
This is line -1.
This is line zero.
This is line one.
This is line -1.
This is line zero.
>This is line one.
Please enter a line:
$Quit
请按任意键继续...
```

#### 4. Conclusion and Discussion

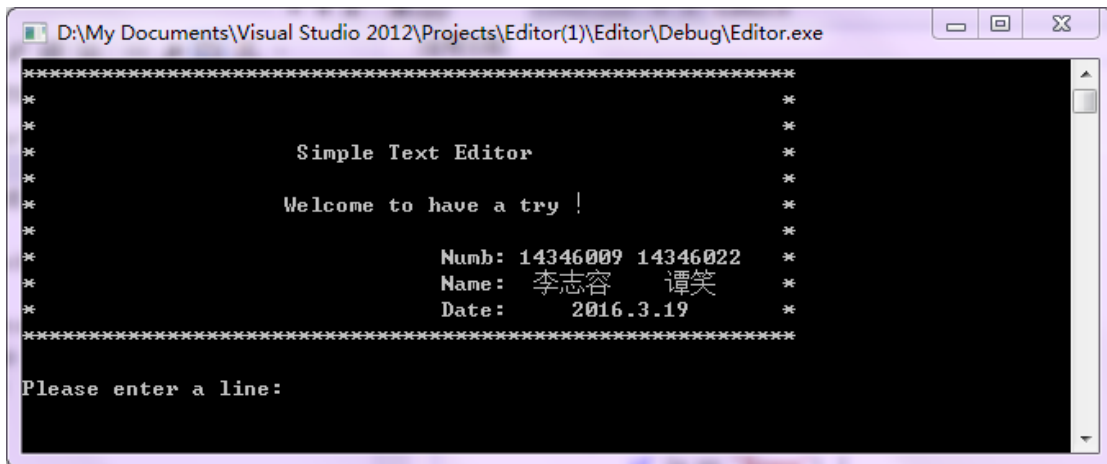
**Achieved:** We have achieved some fundamental functions of an editor like \$Insert, \$Delete, \$Line, \$Print, \$Done, \$Save and \$Open. And they function well.

**Not achieved:** Implementation of some functions may be redundant. It can be more concise.

##### Highlights:

- A. There is a module in “text.cpp” to show our school numbers, names and date:

- B. We have implemented the \$Save and \$Open functions which are not compulsory tasks.



```
*****
*                                                                    *
*                                                                    *
*              Simple Text Editor                                    *
*                                                                    *
*              Welcome to have a try !                              *
*                                                                    *
*              Numb: 14346009 14346022                             *
*              Name: 李志容 谭笑                                    *
*              Date:  2016.3.19                                     *
*                                                                    *
*****

Please enter a line:
```

## 5. Appendices

Refer to the attachment “codes-Editor.pdf”.

## 6. References

- 1) 《数据结构与算法实验实践教程》——乔海燕、蒋爱军、高集荣、刘晓铭
- 2) 《Introduction to Programming with C++》——Y. Daniel Liang
- 3) <http://www.cplusplus.com/reference/list/list/erase/>
- 4) <http://blog.csdn.net/java958199586/article/details/7104315>