```
/*
* 程序包含文本添加，删除，查找，插入统计等功能
* 每次操作前输入操作的选择
*/
#include<iostream>
#include<stdlib.h>
#include<stdio.h>
#include<string>
#include<list>
#include <algorithm>
using namespace std;

/* Cusor 类 */
class Cursor
{
public:
    int line;
    int c;
    Cursor();   // 初始化函数
    void set(int l, int n);     // 设置Cusor属性
    /* 输入和输出的友元类 */
    friend ostream& operator <<(ostream& out, Cursor& a);
    friend istream& operator >>(istream& in, Cursor& a);
};
Cursor::Cursor()
{
    line = 0;
    c = 0;
}
void Cursor::set(int l, int n)
{
```

```
    line = l;
    c = n;
}
ostream& operator <<(ostream& out, Cursor& a)
{
    out<<"(line:"<<a.line<<", char:"<<a.c<<')';
    return out;
}
istream& operator >>(istream& in, Cursor& a)
{

/* TextEditor 类, 实现文本的编辑 */
    cout<<"line:";
    in>>a.line;
    cout<<"char:";
    in>>a.c;
    return in;
}
class TextEditor
{public:
    list<string> article;
    string name;
    Cursor cursor;  // 两个类的聚合关系
    int line;   //总行数

        list<string>::iterator Iterator;

        int total;  //总字数
    TextEditor();
        void mmm();
    string getname();
```

```
    void setname();
    Cursor getcursor();
    void movecursor();   //移动到第L行第C个字符后
    void movecursor(int l, int c);
    void AddText();          //在文档末尾添加文本
    void InsertText();    //在index个字符后插入文本
    void DeleteText();      //删除当前光标前len个字符
    void Findtext();        //查找字符串，并将光标移动到子串开头
        void Conttext();        //统计字符
    void run();
    friend ostream& operator <<(ostream& out, TextEditor& t);
    friend istream& operator >>(istream& in, TextEditor& t);
};
//void TextEditor::mmm(){Iterator=article.iterator;}
TextEditor::TextEditor()
{
    name = "blank";
    line = 0;
    total = 0;
}
string TextEditor::getname()
{
    return name;
}
void TextEditor::setname()
{
    cout<<"设置新的文档名: "<<endl;
    cin>>name;
}
Cursor TextEditor::getcursor()
{
```

```
    return cursor;
}
void TextEditor::movecursor()    //移动到第L行第C个字符后
{
    cin>>cursor;
}
void TextEditor::movecursor(int l, int c)    //移动到第L行第C个字符后
{
    cursor.set(l,c);
}
void TextEditor::AddText()           //在文档末尾添加文本
{
    cout<<"add:";
    string a;
    getchar();
    getline(cin,a);
    if(a!="")            //若输入字符串不为空（不是回车）
    {
        if(line!=0)
        {
            *(article.rbegin())+=a;
        }
        else
        {
            article.push_back(a);
            line++;
        }
        total+=a.length();
    }
    while(a=="")         //若以回车开头
    {
```

```
        getline(cin,a);
        article.push_back(a);
        line++;
        total+=a.length();
    }
    movecursor(line,article.rbegin()->length());      //移动光标到最后
}
void TextEditor::InsertText()    //在当前光标后插入文本
{
    string s;
    if(cursor.line>line)       //输入行数超过总行数，默认为最后一行
        cursor.line = line;
    if(cursor.line<0)
        cursor.line = 1;

    list<string>::iterator i = article.begin();
    for(int j = 1; j<cursor.line; j++,i++);

    if(cursor.c>(*i).length())      //输入字符位置超过串长，默认为最后
        cursor.c = (*i).length();
    if(cursor.c<0)
        cursor.c = 0;
    getchar();
    cout<<"insert:";
    getline(cin,s);
    if(s!="")
        (*i).insert(cursor.c,s);   //在c处插入串s
    while(s=="")    //当输入回车
    {
        string temp;
        temp = (*i).substr(cursor.c+1,(*i).length()-cursor.c);
```

```
        (*i) = (*i).erase(cursor.c+1, (*i).length()-cursor.c);
        i++;
        getline(cin,s);
        s+=temp;
        article.insert(i,s);
        total+=s.length();
    }
    movecursor(line,article.rbegin()->length());      //移动光标到最后
}
void TextEditor::DeleteText()       //删除当前光标前len个字符
{
    if(article.empty())
        return ;

    int len, len_deleted = 0;
    cout<<"detele length:";
    cin>>len;

    list<string>::iterator i = article.begin();
    for(int j = 1; j<cursor.line; i++,j++);

    if(len<0)
        len = 0;

    if(len<=cursor.c)        //删除不超过一行
    {
        (*i) = (*i).erase(cursor.c - len, len);
        len_deleted = len;
    }
    else         //跨行删除
    {
```

```
//先处理当前光标所在行，temp保存剩余字符串
list<string>::iterator de = i;
string temp = (*i).substr(cursor.c,(*i).length()-1);
len_deleted = cursor.c;
i--;
article.remove(*de);
line--;
//循环到删完len个字符或到文档开头
while(len_deleted!=len && i!=(--article.begin()))
{
    if((*i).length() > len-len_deleted)
    {
        (*i) = (*i).erase((*i).length()-(len-len_deleted), len-len_deleted);
        len_deleted = len;
        (*i) += temp;        //将串补入该行
    }
    else
    {
        len_deleted += (*i).length();
        de = i;
        i--;
        article.remove(*de);
        line--;
    }
}
//若删除到文档开头，将剩余的串temp作新行加回
if(i==(--article.begin()))
{
    article.push_front(temp);
    line++;
}
```

```
    }
    total -= len_deleted;
    movecursor(line,article.rbegin()->length());      //移动光标到最后
}
void TextEditor::Findtext()         //查找字符串，并将光标移动到子串开头
{
    string s;
    cout<<"input what you want to find: ";
    cin>>s;
    if(s=="" || s.length()>total)
    {
        movecursor(line,article.rbegin()->length());      //移动光标到最后
        return ;
    }

    list<string>::iterator i = article.begin();
    cursor.line = 1;
    while(i!=article.end())
    {
        cursor.c = (*i).find(s,0);
        if(cursor.c==string::npos)
        {
            i++;
            cursor.line++;
        }
        else
            break;
    }
    if(i==article.end())
        movecursor(line,article.rbegin()->length());
}
```

```
void TextEditor::Conttext()
{
        //cout<<article.front()[0]<<endl;
        int a1=0,a2=0,a3=0,a4=0,a5=0;//字母，数字，标点，空格，总数
        int i;
        for(i=0;i<total;++a5,++i)
        {

                if(article.front()[i]>=65&&article.front()[i]<=90||article.front()
[i]>=97&&article.front()[i]<=122)
                        a1++;
                else if(article.front()[i]>=48&&article.front()[i]<=57)
                        a2++;
                else if(article.front()[i]==32)
                        a4++;
                else
                        a3++;
                        //cout<<article.front()[i]<<endl;
        }
        cout<<"总字符个数为: "<<a5<<endl;
        cout<<"字母个数为: "<<a1<<endl;
        cout<<"数字个数为: "<<a2<<endl;
        cout<<"标点个数为: "<<a3<<endl;
        cout<<"空格个数为: "<<a4<<endl;system("Pause");

}
void TextEditor::run()
{
    char chose = ' ';
    while(chose!='q')
    {
```

```
        cout<<"文档名: "<<name<<endl;
        cout<<*this;
        cout<<"----------------------"<<endl;
        cout<<"当前参数: ";
        cout<<"cursor: "<<cursor<<endl;
        cout<<"line:"<<line<<endl;
        cout<<"length:"<<total<<endl;
        cout<<"功能: \n";
        cout<<"1: AddText;\n2: DeleteText; \n3: FindText; \n4: InsertText;\n5:
MoveCursor;\n6: SetName; \n7: Count;\n8: Quit; "<<endl;
        cout<<"输入您想进行的操作: "<<endl;
        cin>>chose;
        switch(chose)
        {
            case '1': AddText(); break;
            case '4': InsertText(); break;
            case '2': DeleteText(); break;
            case '3': Findtext(); break;
            case '5': movecursor(); break;
            case '6': setname(); break;
                        case '7': Conttext();break;
                        case '8': return;
            default :  break;
        }
        system("cls");
    }
}
ostream& operator <<(ostream& out, TextEditor& t)
{
    list<string>::iterator i = t.article.begin();
    for( ; i != t.article.end(); i++)
```

```
    {
        out<<(*i)<<endl;
    }
    return out;
}

int main()
{
    TextEditor test;
    test.run();

        system("Pause");
        return 0;
}
```