

Міністерство освіти і науки України  
Національний технічний університет України «Київський  
політехнічний інститут імені Ігоря Сікорського»  
Факультет Інформатики та обчислювальної  
Техніки Кафедра інформатики та програмної інженерії

Звіт  
з лабораторної роботи № 1.1 з дисципліни  
«Основи програмування - 2.  
Модульне програмування.»

“Текстові файли”  
Варіант: 30

Виконав студент ІП-11 Тихонов Федір Сергійович  
(шифр, прізвище, ім'я, по батькові)

Перевірив  
(прізвище, ім'я, по батькові)

# Лабораторна робота №1.1

## Текстові файли.

**Мета** - вивчити особливості створення і обробки файлів даних.

### Постановка задачі:

Створити текстовий файл. Сформувати новий текстовий файл, що складається з рядків вихідного файлу, розміщених у порядку спадання кількості символів у рядку. На початку кожного рядка нового файлу вказати кількість символів у рядку. Вивести вміст вхідного і вихідного файлів.

### Розв'язання на C++:

#### 1) Header

```
#pragma once
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
using namespace std;

void pushLinesInFile(const string& fileName);
//push lines into file

void outputContents(const string& oldFileName, const string&
newFileName); //output contents of the file

vector<string> getLines(const string& fileName); //copy lines from
file to vector

void copyLinesToAnotherFile(const string& fileName1, string
fileName2); //copy lines from file1 to file2

int countChars(const string& line); //count characters in a string

void sortStrings(vector<string>& lines); //sort strings in
vector

void DeleteFile(const string& oldFileName, const string& newFileName,
const string& permission); //BASED on th decision made by the user,

// delete contents of files or not.
string askForPermission();
```

#### 2) .cpp Library file

```
#include "lib.h"
```

```

using namespace std;

void pushLinesInFile(const string& fileName) {
    ofstream file;
    file.open(fileName, ios::app);
    string line;
    cout << "Enter lines." << endl;
    while (getline(cin, line)) {
        if (!line.empty()) {
            file << line << endl;
        }
    }
    file.close();
}

void outputContents(const string& oldFileName, const string&
newFileName){
    string line;
    ifstream oldFile;
    ifstream newFile;
    oldFile.open(oldFileName, ios::in);
    newFile.open(newFileName, ios::in);
    cout << endl << "Outputting contents for oldFile:" << endl;
    while (getline(oldFile, line)) {
        cout << line << endl;
    }
    cout << endl << "Outputting contents for newFile:" << endl;
    while (getline(newFile, line)) {
        cout << line << endl;
    }
    oldFile.close();
    newFile.close();
}

void DeleteFile(const string& oldFileName, const string& newFileName,
const string& permission){
    if(permission == "no"){
        ofstream oldFile(oldFileName, ios::out);
        ofstream newFile(newFileName, ios::out);
        oldFile.close();
        newFile.close();
    }
}

void sortStrings(vector<string>& lines){
    string buffer;
    for(int i = 0; i<lines.size(); i++){
        for(int j = 0; j< lines.size() - i - 1; j++){
            if(countChars(lines[j]) < countChars(lines[j+1])){
                buffer = lines[j];
                lines[j] = lines[j+1];
                lines[j+1] = buffer;
            }
        }
    }
}

```

```

}
int countChars(const string& line){
    int count = 0;
    for(char i : line){
        if(i != ' '){
            count++;
        }
    }
    return count;
}

vector<string> getLines(const string& fileName){
    string line;
    ifstream file(fileName, ios::in);
    vector<string> lines;
    while(!file.eof()){
        getline(file, line);
        lines.push_back(line);
    }
    file.close();
    return lines;
}

void copyLinesToAnotherFile(const string& fileName1, string
fileName2){
    ifstream file1(fileName1, ios::in);
    ofstream file2(fileName2, ios::out);
    vector<string> lines = getLines(fileName1);
    sortStrings(lines);
    for(const string& line : lines){
        if(!line.empty()) {
            file2 << "Length of this line is " << countChars(line) <<
"----" << line << endl;
        }
    }
    file1.close();
    file2.close();
}

string askForPermission(){
    string answer;
    cout << "Do you want to save your work? Type 'no' to delete it
afterwards." << endl;
    getline(cin, answer);
    return answer;
}

```

### 3) main

```

#include "lib.h"
using namespace std;
int main(){

    string oldFileName = "oldFile.txt"; //naming our files
    string newFileName = "new.txt";

```

```
    string perm = askForPermission(); //asking the user whether to  
    save their text.
```

```
    pushLinesInFile(oldFileName); //push lines into a source file.
```

```
    copyLinesToAnotherFile(oldFileName, newFileName); //copy lines  
    from file1 to file2,
```

```
    //sort lines.
```

```
    outputContents(oldFileName, newFileName);  
    //output everything in our files to console.
```

```
    DeleteFile(oldFileName, newFileName, perm);  
    //BASED on th decision made by the user,
```

```
    // delete contents of files or not.
```

```
    return 0;  
}
```

**Результати на C++:**

Do you want to save your work? Type 'no' to delete it afterwards.

Enter lines.

orepgn

efpnfep[nef

efpoef[ojfe

feobfepobfepoibefbpi

efpief

efpifepeifnfpfien

^D

Outputting contents for oldFile:

orepgn

efpnfep[nef

efpoef[ojfe

feobfepobfepoibefbpi

efpief

efpifepeifnfpfien

Outputting contents for newFile:

Length of this line is 20----feobfepobfepoibefbpi

Length of this line is 16----efpifepeifnfpfien

Length of this line is 11----efpnfep[nef

Length of this line is 11----efpoef[ojfe

Length of this line is 6----orepgn

Length of this line is 6----efpief

Process finished with exit code 0

```
CMakeLists.txt × main.cpp × lib.cpp × new.txt × oldFile.txt × lib.h ×
1 Length of this line is 20----feobfepobfepoibefbpi
2 Length of this line is 16----efpifepeifnfpfien
3 Length of this line is 11----efpnfep[nef
4 Length of this line is 11----efpoef[ojfe
5 Length of this line is 6----orepgn
6 Length of this line is 6----efpief
```

```
CMakeLists.txt × main.cpp × lib.cpp × new.txt × oldFile.txt × lib.h ×
1 orepgn
2 eonfep[nef
3 efpoef[ojfe
4 feobfepobfepoibefbpi
5 efpief
6 efpifepeifnfpfien
```

Do you want to save your work? Type 'no' to delete it afterwards.

no

Enter lines.

pinefip

feibefpb

efpiefpfbeipectb

^D

Outputting contents for oldFile:

feionfep

fepibefp[nfe

e

pinefip

feibefpb

efpiefpfbeipectb

Outputting contents for newFile:

Length of this line is 15----efpiefpfbeipectb

Length of this line is 12----fepibefp[nfe

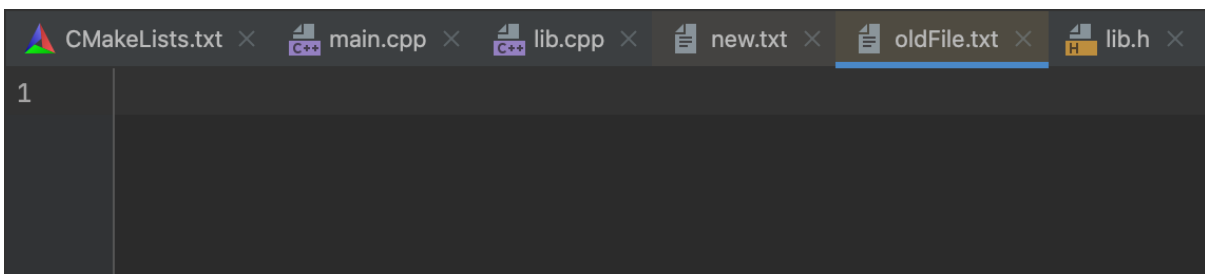
Length of this line is 8----feionfep

Length of this line is 8----feibefpb

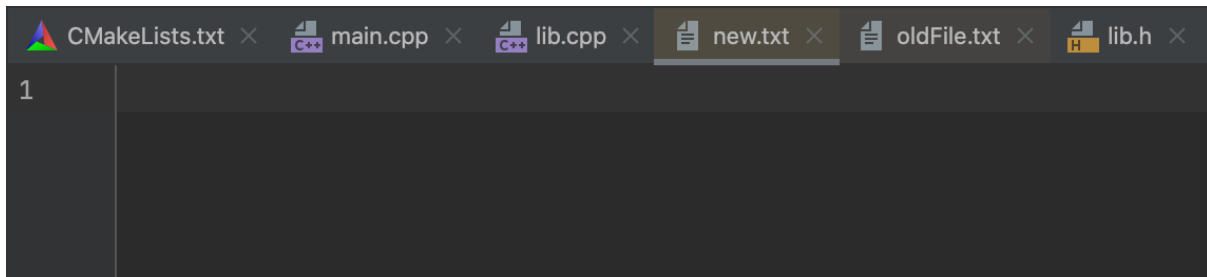
Length of this line is 7----pinefip

Length of this line is 1----e

Process finished with exit code 0







Код на Python:

python module:

```
def sortStrings(StringList):  
    for i in range(len(StringList)):  
        for j in range(len(StringList) - i - 1):  
            if countChars(StringList[j]) < countChars(StringList[j + 1]):  
                StringList[j], StringList[j + 1] = StringList[j + 1],  
StringList[j]  
    return StringList
```

```
def countChars(line):  
    i = 0  
    for char in line:  
        if char != ' ':  
            i += 1  
    return i
```

```
def fileToList(fileName):  
    lineList = []  
    with open(fileName, 'r') as file:  
        for line in file:  
            lineList.append(line)  
    return lineList
```

```
def pushLinesToFile(fileName):  
    with open(fileName, 'a') as file:  
        answer = ""  
        line = ""  
        while answer != "STOP":  
            line = input("Enter a line:\n")  
            file.write(line + "\n")  
            answer = input("If you are done writing, type \"STOP\"\n")  
    return None
```

```
def listToFile(lineList, newFileName):  
    with open(newFileName, 'w') as file:  
        for line in lineList:  
            newLine = "The length of this line equals to " +  
str(countChars(line)) + " " + line  
            file.write(newLine)
```

```
return None
```

```
def askToDelete(sourceFileName, newFileName):  
    answer = input("Do you want to save your work?\nType \"DO\" not to.")  
    if answer == "DO":  
        sourceFile = open(sourceFileName, 'w')  
        sourceFile.close()  
        newFile = open(newFileName, 'w')  
        newFile.close()  
        print("New File and Source File have been deleted.\n")  
    return None
```

```
def outputFiles(sourceFileName, newFileName):  
    with open(sourceFileName, 'r') as f:  
        print(f.read())  
    with open(newFileName, 'r') as f:  
        print(f.read())  
    return None
```

**main file:**

```
import Lib_Lab1
```

```
sourceFileName = "sourceFile.txt"          # Names of files.  
newFileName = "newFile.txt"
```

```
Lib_Lab1.pushLinesToFile(sourceFileName)    # Push text to source  
file.
```

```
lineList = Lib_Lab1.fileToList(sourceFileName) # Copy contents of the  
file to list.
```

```
lineList = Lib_Lab1.sortStrings(lineList)    # Sort list.
```

```
Lib_Lab1.listToFile(lineList, newFileName)   # Copy contents of the  
sorted list to a new file.
```

```
Lib_Lab1.askToDelete(sourceFileName, newFileName) # Leave it up to user  
to decide whether to save the file or not.
```

**Результати на Python:**

```
Enter a line:
If you are done writing, type "STOP"
Enter a line:
If you are done writing, type "STOP"
Enter a line:
If you are done writing, type "STOP"
Enter a line:
If you are done writing, type "STOP"
elfjb eflkbflkeb
efbepfbefpbefpibef
pefbepfbepbfepbepfbe
epfibepfbefpbef

The length of this line equals to 21 pefbepfbepbfepbepfbe
The length of this line equals to 19 efbepfbefpbefpibef
The length of this line equals to 16 elfjb eflkbflkeb
The length of this line equals to 16 epfibepfbefpbef

Do you want to save your work?
Type "D0" not to.
Process finished with exit code 0
```

### **Висновок:**

Отже, ми навчилися використовувати текстові файли для роботи з даними. Файли надають більше можливостей, оскільки завдяки ним можна зберігати дані вже після того, як програма завершила роботу.