

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 з дисципліни

«Основи програмування»

«Робота з текстовими файлами»

Варіант 11

Виконав студент ІП-13, Дем'янчук Олександр Петрович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 202 1

Лабораторна робота № 1

Тема: Робота з текстовими файлами

Мета: вивчити особливості створення і обробки текстових файлів даних.

Варіант 11

Завдання: Створити текстовий файл. Визначити в кожному його рядку кількість слів і довжину найбільшого слова. Дописати значення кількості слів на початок відповідного рядка, довжину максимального слова - в його кінець, Вивести вміст вхідного і перетвореного файлів.

Постановка задачі: створити текстовий файл. Переписати його компоненти до нового текстового файлу, дописуючи на початку кожного рядка значення кількості слів, а в кінець рядка - довжину максимального слова. Вивести вміст вихідного і створеного файлів.

Код C++

main.cpp

```
#include "func.h"
```

```
int main()
{
    setlocale(LC_ALL, ".1251");
    string oldPath = "old.txt";
    string newPath = "new.txt";
    createFile(oldPath);
    createNew(oldPath, newPath);
    cout << "your file : \n";
    outputFile(oldPath);
    cout << "new file : \n";
    outputFile(newPath);
}
```

func.cpp

```
#include "func.h"
```

```
#include "func.h"
```

```
void createFile(string path) {
    ofstream outfile(path, ofstream::app);
    if (outfile) {
        cout << "ENTER to end line.\n CTRL+Z - end input file.\n";
        string line;
        getline(cin, line, '\26');
        outfile << line;
        outfile.close();
    }
    else {
        cout << "a file can't be opened";
    }
}
```

```
void outputFile(string path) {
    ifstream inFile;
    inFile.open(path);
    while (!inFile.eof()) {
        string textLine;
        getline(inFile, textLine);
        cout << textLine << endl;
    }
    inFile.close();
}
```

```
vector<string> split(string line, char sep = ' ') {
    vector<string> res;
    string slice = "";
    line += sep;
    for (int i = 0; i < int(line.length()); i++) {
        if (line[i] == sep) {
            if (slice.length() > 0) res.push_back(slice);
            slice = "";
        }
        else slice += line[i];
    }
    return res;
}
```

```
int wordAmt(string line) {
```

```

    vector<string> array = split(line);
    return array.size();
}

int maxLength(string line) {
    vector<string> array = split(line);
    int max = array[0].length();
    for (size_t i = 0; i < array.size(); i++){
        if (array[i].length() > max) {
            max = array[i].length();
        }
    }
    return max;
}

string transform(string line) {
    return "[" + to_string(wordAmt(line)) + "]" + line + " [" +
to_string(maxLength(line)) + "];"
}

void createNew(string oldPath, string newPath) {
    ifstream oldFile;
    ofstream newFile;
    oldFile.open(oldPath);
    newFile.open(newPath);
    if (!oldFile.is_open() || !newFile.is_open()) {
        cout << "a file can't be opened" << endl;
    }
    else {
        while (!oldFile.eof()) {
            string temp = "";
            getline(oldFile, temp);
            if (temp.length() > 0) {
                temp = transform(temp);
                newFile << temp << "\n";
            }
        }
        oldFile.close();
        newFile.close();
    }
}

```

func.h

```
#pragma once
#include <iostream>
#include <string>
#include <fstream>
#include <vector>

using namespace std;

void createFile(string);
void outputFile(string);
vector<string> split(string, char);
int wordAmt(string);
int maxLength(string);
string transform(string);
void createNew(string, string);
```

Код python

lab1py.py

```
from module1 import *

oldpath = "old.txt"
newpath = "new.txt"
createfile(oldpath)
createnew(oldpath, newpath)
print("your file: ")
outputfile(oldpath)
print("new file: ")
outputfile(newpath)
```

module1.py

```
def createfile(file):
    print("ENTER to end line.\n >< - end input file.\n")
    lines = []
    while True:
        line = input()
        if line.find("><") != -1:
            lines.append(line[0:-2])
            break
        else:
            lines.append(line)
    file = open(file, "a")
    file.write('\n'.join(lines))
    file.close()

def outputfile(path):
    file = open(path, 'r')
    for line in file:
        print(line)
    file.close()

def wordamt(line):
    return len(line.split())

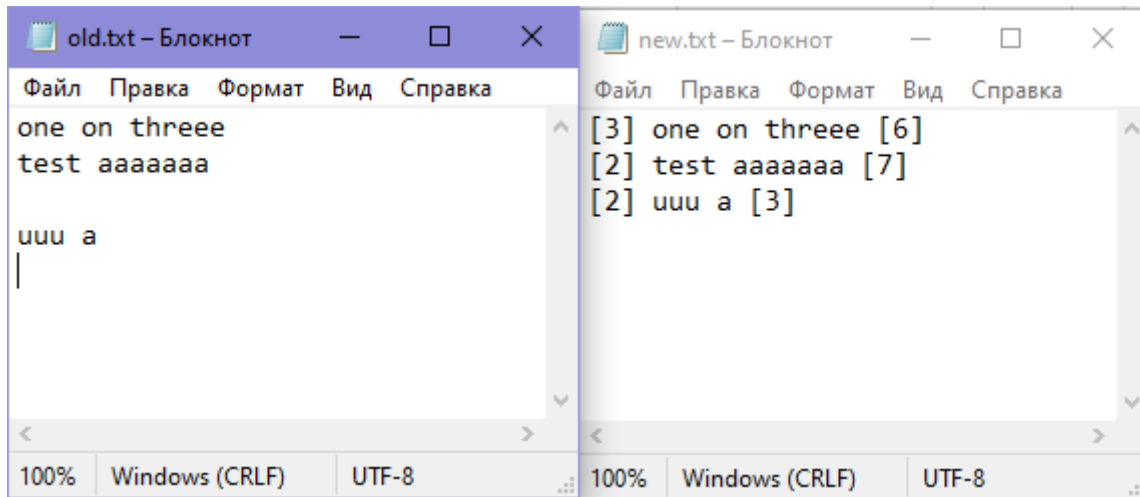
def maxlength(line):
    words = line.split()
    max = 0
    for i in range(len(words)):
        if max < len(words[i]):
            max = len(words[i])
    return max

def transform(line):
    if len(line) > 0:
        return "[" + str(wordamt(line)) + "]" + line + "[" + str(maxlength(line)) + "]\n"
    else:
        return ""

def createnew(oldpath, newpath):
    old = open(oldpath, 'r')
    new = open(newpath, 'w')
    for line in old:
        line = line.rstrip(line[-1])
        line = transform(line)
        new.write(line)
    old.close()
    new.close()
```

Результат роботи

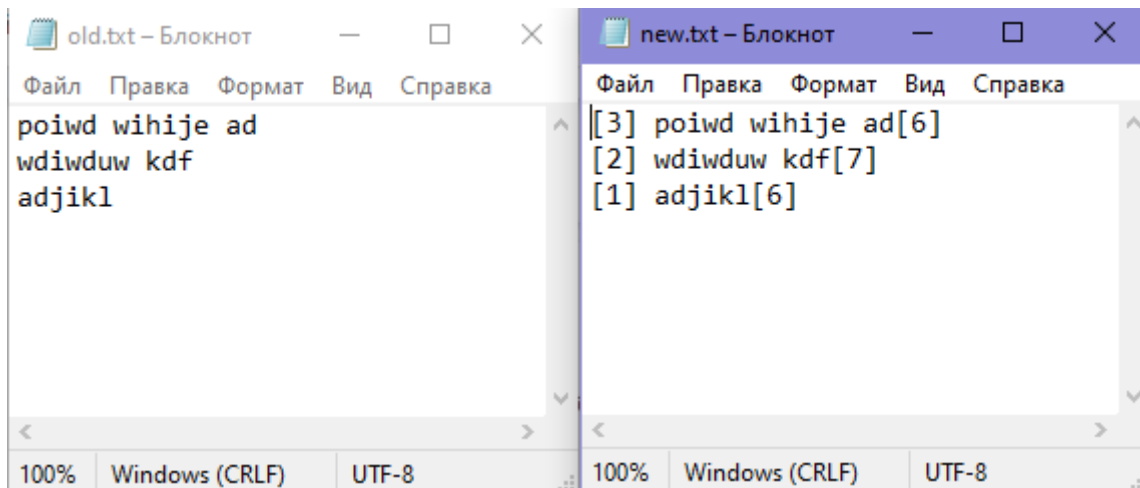
C++



The screenshot shows two Notepad++ windows side-by-side. The left window, titled 'old.txt - Блокнот', contains the text:
one on threee
test aaaaaaa

uuu a
|
The right window, titled 'new.txt - Блокнот', contains the output of a C++ program:
[3] one on threee [6]
[2] test aaaaaaa [7]
[2] uuu a [3]
Both windows show a status bar at the bottom with '100%', 'Windows (CRLF)', and 'UTF-8'.

Python



The screenshot shows two Notepad++ windows side-by-side. The left window, titled 'old.txt - Блокнот', contains the text:
poiwd wihije ad
wdiwduw kdf
adjikl
The right window, titled 'new.txt - Блокнот', contains the output of a Python program:
[3] poiwd wihije ad[6]
[2] wdiwduw kdf[7]
[1] adjikl[6]
Both windows show a status bar at the bottom with '100%', 'Windows (CRLF)', and 'UTF-8'.