

Міністерство освіти і науки України
КПІ ім. Ігоря Сікорського
Кафедра ІІІ

ЗВІТ
з виконання лабораторної роботи № 1
з кредитного модуля
“Основи програмування-2. Методології програмування”

Варіант № 21

Виконав:
студент 1-го курсу
гр. ІІ-22 ФІОТ
Патріюк Юрій Олексійович

Київ 2023

ПОСТАНОВКА ЗАДАЧІ

21. Створити текстовий файл. Слова у тексті відділені пробілами, комами, крапками; групи слів можуть бути обмежені фігурними дужками. Перевірити, чи є баланс дужок у кожному рядку даного тексту (без вкладеності дужок): для кожної дужки, яка відкривається, справа має бути дужка, що закривається. Рядки вхідного файлу, у яких порушений баланс дужок, переписати у новий текстовий файл. Відкоригувати баланс дужок у новому файлі шляхом видалення зайвих дужок. У кожному фрагменті тексту, обмеженому парою дужок (що залишилися), визначити кількість слів тексту, які містять хоча б одну цифру. Вставити цю величину на початок фрагменту (після відкриваючої дужки), відділивши її символом «-». Вивести вміст вхідного і створеного файлів.

ТЕКСТ ПРОГРАМИ

Вміст файлу main.cpp

```
#include <iostream>
#include "functions.h"
#include "functions_c.h"

int main(int argc, char *argv[]) {
    string name = "main.txt";
    string secondName = "to_correct.txt";

    if(argc==3 && strcmp(argv[1], "-mode")==0) {
        if(strcmp(argv[2], "FilePointer")==0)
        {
            WriteInFile_C(name.c_str());
            ShowFile_C(name.c_str());
            AddText_C(name.c_str());
            FindAndCopy_C(name.c_str());
            ShowFile_C(secondName.c_str());
            ClearFileFromExcessiveBraces_C(secondName.c_str());
            ShowFile_C(secondName.c_str());
            FindNumberInBraces_C(secondName.c_str());
            cout<<"-----Results-----"<<endl;
            ShowFile_C(name.c_str());
            ShowFile_C(secondName.c_str());
        }
        else if(strcmp(argv[2], "FileStream")==0)
        {
            WriteInFile(name);
            ShowFile(name);
            AddText(name);
            FindAndCopy(name);
            ShowFile(secondName);
            ClearFileFromExcessiveBraces(secondName);
            ShowFile(secondName);
        }
    }
}
```

```

        FindNumberInBraces(secondName);
        cout<<"-----Results-----"<<endl;
        ShowFile(name);
        ShowFile(secondName);
    }
}

cout<<"-----Finish-----"<<endl;
system("pause");
return 0;
}

```

Вміст файлу functions.h

```

using namespace std;

void ShowFile(const string&);
void WriteInFile(const string&);
void AddText(const string&);
void FindAndCopy(const string&);
void ClearFileFromExcessiveBraces(const string&);
void FindNumberInBraces(const string&);

```

Вміст файлу functions_c.h

```

#include <cstring>
using namespace std;

void ShowFile_C(const char*);
void AddText_C(const char*);
void WriteInFile_C(const char*);
void FindAndCopy_C(const char*);
void ClearFileFromExcessiveBraces_C(const char*);
void FindNumberInBraces_C(const char*);

```

Вміст файлу functions.cpp

```

#include <iostream>
#include "functions.h"
#include <fstream>

//Обрахунок кількості слів з числами
int WordsWithNumbers(const string& str)
{
    int counter = 0;
    bool numCheck = false;
    for(int i = 0; i<str.length(); i++)
    {
        if(isdigit(str[i]))
            numCheck = true;
        if((str[i] == ' ' || i==str.length()-1) && numCheck)
        {
            counter++;
            numCheck=false;
        }
    }

    return counter;
}

//Вивід вмісту файлу

```

```

void ShowFile(const string& name){
    ifstream iFile(name);
    string line;
    cout<<"Content of "<<name<<endl;
    while(!iFile.eof())
    {
        getline(iFile,line);
        cout<<'<<line<<endl;
    }
    cout<<endl;
    iFile.close();
}
//Додання нових рядків
void AddText(const string& name)
{
    char choose;
    cout<<"Do you want to add lines? y/n:";
    cin>>choose;
    if(choose == 'y')
    {
        ofstream outFile;
        outFile.open(name,ios::app);
        cout<<"Click double enter to finish writing"<<endl;
        string line=" ";
        fflush(stdin);
        while(!line.empty()) {
            cout<<"#";
            getline(cin, line);
            if(!line.empty())
                outFile << line << endl;
        }
        outFile.close();
        cout<<endl;
        ShowFile(name);
    }
}
//Запис у файл
void WriteInFile(const string& name)
{
    ofstream outFile(name);
    cout<<"Click double enter to finish writing"<<endl;
    string line=" ";
    while(!line.empty()) {
        cout<<"#";
        getline(cin, line);
        if(!line.empty())
            outFile << line << endl;
    }
    outFile.close();
}
//Пошук неправильних рядків та запис у окремий файл
void FindAndCopy(const string& name)
{
    ifstream inFile(name);
    ofstream additionalFile("to_correct.txt");
    string line;
    while(!inFile.eof())
    {
        bool is_incorrect = false;
        bool is_opened = false;
        getline(inFile,line);
        if(!line.empty()) {
            int i = -1;

```

```

        int size = (int) line.length();
        while (!is_incorrect && i < size) {
            ++i;
            if (line[i] != '{' && line[i] != '}')
                continue;
            if (line[i] == '{' && !is_opened)
                is_opened = true;
            else if (line[i] == '{' && is_opened || line[i] == '}' && !is_opened)
                is_incorrect = true;
            else if (line[i] == '}' && is_opened)
                is_opened = false;
        }
        if (is_incorrect || is_opened)
            additionalFile << line << endl;
    }
}
inFile.close();
additionalFile.close();
}
//зачистка рядка від зайвих фігурних дужок
void ClearFileFromExcessiveBraces(const string& name)
{
    ifstream iFile(name);
    string oFileName = "temp.txt";
    ofstream oFile(oFileName);
    string line;

    while(!iFile.eof()) {
        getline(iFile, line);
        if(!line.empty()) {
            bool is_opened = false;
            int i = 0,
                open_position = -1,
                size = (int) line.length();
            while (i < size) {
                if (line[i] == '{' && !is_opened) {
                    is_opened = true;
                    open_position = i;
                } else if (line[i] == '{' && is_opened) {
                    line.erase(open_position, 1);
                    size = (int) line.length();
                    is_opened = false;
                    i -= 2;
                } else if (line[i] == '}' && !is_opened) {
                    line.erase(i, 1);
                    size = (int) line.length();
                    i--;
                } else if (line[i] == '}' && is_opened)
                    is_opened = false;
                ++i;
            }
            if (is_opened)
                line.erase(open_position);
            oFile << line << endl;
        }
    }

    iFile.close();
    remove(name.c_str());
    oFile.close();
    rename(oFileName.c_str(), name.c_str());
}
//пошук слів з числами та запис кількості у фігурних дужках
void FindNumberInBraces(const string& name)

```

```

{
    ifstream iFile(name);
    string oFileName = "temp.txt";
    ofstream oFile(oFileName);
    string line;

    while(!iFile.eof()) {
        getline(iFile, line);
        if(!line.empty()) {
            int open_position = 0, close_position = 0;
            while (open_position != string::npos && close_position != string::npos) {
                open_position = (int) line.find('{', open_position);
                close_position = (int) line.find('}', close_position);

                if (open_position != string::npos && close_position != string::npos)
                {
                    string temp = line.substr(open_position + 1, close_position -
open_position - 1);
                    int amount = WordsWithNumbers(temp);
                    if (amount != 0) {
                        line.insert(open_position + 1, '-' + to_string(amount) + '-'
');
                        close_position += 4;
                    } else
                        close_position++;
                    open_position = close_position;
                }
            }
            oFile << line << endl;
        }
    }
    iFile.close();
    remove(name.c_str());
    oFile.close();
    rename(oFileName.c_str(),name.c_str());
}

```

Вміст файлу functions_c.cpp

```

#include <iostream>
#include "functions_c.h"
#define MAX_LINE_LENGTH 500

//видалення елемента за індексом
void RemoveElementByIndex(char* array,int index)
{
    if(index<0)
        index = 0;
    while(array[index]!=0)
    {
        array[index] = array[index+1];
        index++;
    }
}

//пошук позиції вказаного символу
int FindPosition(const char* array, int start_position,char elem)
{
    if(strlen(array)<start_position)
        return -1;
    else {
        int i = start_position;
        while (array[i] != elem && array[i] != '\n')
            i++;
    }
}

```

```

        if (array[i] == elem)
            return i;
        else
            return -1;
    }
}
//копіювання масиву з позиції довжиною length
void Copy(const char* array, char* buff, int start_position, int length)
{
    int i = 0;
    while(i<length)
    {
        buff[i] = array[start_position+i];
        i++;
    }
}
//Обрахунок кількості слів з числами
int WordsWithNumbers(const char* str)
{
    int counter = 0;
    bool numCheck = false;
    for(int i = 0; i<strlen(str); i++)
    {
        if(isdigit(str[i]))
            numCheck = true;
        if((str[i] == ' ' || i == strlen(str)-1) && numCheck)
        {
            counter++;
            numCheck=false;
        }
    }
    return counter;
}
//вставлення на позицію кількості слів з числами
void InsertAmount(char* array, int position, int amount)
{
    int i = -1;
    char result[MAX_LINE_LENGTH] = "";
    char temp[MAX_LINE_LENGTH]="";
    Copy(array, temp,position, (int)strlen(array)-position);
    while (i!=position)
    {
        i++;
        if(i==position)
        {
            strcat(result, ("-"+to_string(amount)+"-").c_str());
        }
        else {
            result[i] = array[i];
        }
    }

    strcat(result,temp);
    strcpy(array,result);
}

//Вивід вмісту файлу
void ShowFile_C(const char* name)
{
    FILE* file;
    file = fopen(name,"r");

```

```

char line[MAX_LINE_LENGTH];
cout<<"Content of "<<name<<endl;
while(fgets(line,MAX_LINE_LENGTH,file))
{
    cout<<'# '<<line;
}
cout<<endl;
fclose(file);
}
//Додавання нових рядків
void AddText_C(const char* name)
{
    char choose;
    cout<<"Do you want to add lines? y/n:";
    cin>>choose;
    if(choose == 'y')
    {
        FILE* file;
        file = fopen(name,"a");
        cout<<"Click double enter to finish writing"<<endl;
        char line[MAX_LINE_LENGTH] = " ";
        fflush(stdin);
        while(strcmp(line,"")!=0) {
            cout<<"# ";
            gets(line);
            if(strcmp(line,"")!=0) {
                fputs(line, file);
                fputs("\n", file);
            }
        }
        fclose(file);
        cout<<endl;
        ShowFile_C(name);
    }
}
//Запис у файл
void WriteInFile_C(const char* name)
{
    FILE* file;
    file = fopen(name,"w");
    char line[MAX_LINE_LENGTH] = " ";
    cout<<"Click double enter to finish writing"<<endl;
    while(strcmp(line,"")!=0) {
        cout<<"# ";
        gets(line);
        if(strcmp(line,"")!=0) {
            fputs(line, file);
            fputs("\n", file);
        }
    }
    fclose(file);
}
//Пошук неправильних рядків та запис у окремий файл
void FindAndCopy_C(const char* name)
{
    FILE* main, *to_correct;
    main = fopen(name,"r");
    to_correct = fopen("to_correct.txt","w");
    char line[MAX_LINE_LENGTH] = " ";
    while(fgets(line,MAX_LINE_LENGTH,main))
    {
        bool is_incorrect = false;
        bool is_opened = false;
    }
}

```



```

    int i = -1;
    while (!is_incorrect && line[i]!='\n') {
        ++i;
        if (line[i] != '{' && line[i] != '}')
            continue;
        if (line[i] == '{' && !is_opened)
            is_opened = true;
        else if (line[i] == '{' && is_opened || line[i] == '}' && !is_opened)
            is_incorrect = true;
        else if (line[i] == '}' && is_opened)
            is_opened = false;
    }
    if (is_incorrect || is_opened)
        fputs(line,to_correct);

}
fclose(main);
fclose(to_correct);
}
//зачистка рядка від зайвих фігурних дужок
void ClearFileFromExcessiveBraces_C(const char* name)
{
    FILE* oFile, *tFile;
    oFile = fopen(name,"r");
    char oFileName[] = "temp.txt";
    tFile = fopen(oFileName,"w");
    char line[MAX_LINE_LENGTH] = " ";

    while(fgets(line,MAX_LINE_LENGTH,oFile)) {
        if(strcmp(line,"")!=0) {
            bool is_opened = false;
            int i = 0,
                open_position = -1;
            while (line[i] != '\n') {
                if (line[i] == '{' && !is_opened) {
                    is_opened = true;
                    open_position = i;
                } else if (line[i] == '{' && is_opened) {
                    RemoveElementByIndex(line,open_position);
                    is_opened = false;
                    i -= 2;
                } else if (line[i] == '}' && !is_opened) {
                    RemoveElementByIndex(line,i-1);
                    i--;
                } else if (line[i] == '}' && is_opened)
                    is_opened = false;
                ++i;
            }
            if (is_opened)
                RemoveElementByIndex(line,open_position);
            fputs(line, tFile);
            memset(line,0,sizeof(line));
        }
    }

    fclose(oFile);
    remove(name);
    fclose(tFile);
    rename(oFileName,name);
}
//пошук слів з числами та запис кількості у фігурних дужках
void FindNumberInBraces_C(const char* name)
{
    FILE* oFile, *tFile;

```

```

oFile = fopen(name,"r");
char oFileName[] = "temp.txt";
tFile = fopen(oFileName,"w");
char line[MAX_LINE_LENGTH] = " ";

while(fgets(line,MAX_LINE_LENGTH,oFile)) {
    if(strcmp(line,"")!=0) {
        int open_position = 0, close_position = 0;
        while (open_position != -1 && close_position != -1) {
            open_position = FindPosition(line,open_position,'{');
            close_position = FindPosition(line,close_position,'}');

            if (open_position != -1 && close_position != -1) {
                char temp[MAX_LINE_LENGTH]="";
                Copy(line,temp,open_position+1,close_position-open_position-1);
                int amount = WordsWithNumbers(temp);
                if (amount != 0) {
                    InsertAmount(line,open_position+1,amount);
                    close_position += 4;
                } else
                    close_position++;
                open_position = close_position;
            }
        }
        fputs(line,tFile);
    }
}
fclose(oFile);
remove(name);
fclose(tFile);
rename(oFileName,name);
}

```

РЕЗУЛЬТАТ ТЕСТУВАННЯ

```
start Lab1.exe -mode FileStream
```

```
Click double enter to finish writing
#{{{word}}}}
#{{test1 test2}}
#check
#{lorem ipsum}
#{{line1 line2 line3 line4}}
#
Content of main.txt
#{{{word}}}}
#{{test1 test2}}
#check
#{lorem ipsum}
#{{line1 line2 line3 line4}}
#
Do you want to add lines? y/n:y
Click double enter to finish writing
#{{word1 word2 word3}}
#{{i want2 to believe4}}
#
Content of main.txt
#{{{word}}}}
#{{test1 test2}}
#check
#{lorem ipsum}
#{{line1 line2 line3 line4}}
#{{word1 word2 word3}}
#{{i want2 to believe4}}
#
Content of to_correct.txt
#{{{word}}}}
#{{test1 test2}}
#{{line1 line2 line3 line4}}
#{{word1 word2 word3}}
#{{i want2 to believe4}}
#
Content of to_correct.txt
#{word}
#{test1 test2}
#{{line1 line2 line3 line4}}
#{word1 word2 word3}
#{i want2 to believe4}
#
-----Results-----
Content of main.txt
#{{{word}}}}
#{{test1 test2}}
#check
#{lorem ipsum}
#{{line1 line2 line3 line4}}
#{{word1 word2 word3}}
#{{i want2 to believe4}}
#
Content of to_correct.txt
#{word}
#{-2-test1 test2}
#{{-4-line1 line2 line3 line4}}
#{-3-word1 word2 word3}
#{-2-i want2 to believe4}
#
-----Finish-----
Press any key to continue . . .
```

```
start Lab1.exe -mode FilePointer
```

```
Click double enter to finish writing
#{lorem ipsum}
#{{word1 word2 word3}}
#{{test1 test2}}
#check
#
Content of main.txt
#{lorem ipsum}
#{{word1 word2 word3}}
#{{test1 test2}}
#check
Do you want to add lines? y/n:y
Click double enter to finish writing
#{{{word}}}}
#{{line1 line2 line3 line4}}
#
Content of main.txt
#{lorem ipsum}
#{{word1 word2 word3}}
#{{test1 test2}}
#check
#{{{word}}}}
#{{line1 line2 line3 line4}}
#
Content of to_correct.txt
#{{word1 word2 word3}}
#{{test1 test2}}
#{{{word}}}}
#{{line1 line2 line3 line4}}
#
Content of to_correct.txt
#{word1 word2 word3}
#{test1 test2}
#{word}
#{{line1 line2 line3 line4}}
#
-----Results-----
Content of main.txt
#{lorem ipsum}
#{{word1 word2 word3}}
#{{test1 test2}}
#check
#{{{word}}}}
#{{line1 line2 line3 line4}}
#
Content of to_correct.txt
#{-3-word1 word2 word3}
#{-2-test1 test2}
#{word}
#{{-4-line1 line2 line3 line4}}
#
-----Finish-----
Press any key to continue . . .
```

Посилання на GitHub: <https://github.com/YuriiPatriuk12/Labs.git>