

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 № 1

Виконав:

студент 1-го курсу

гр. ІІІ-22 ФІОТ

Андреева Уляна Андріївна

Київ 2023

ПОСТАНОВКА ЗАДАЧІ

Створити текстовий файл, що представляє собою фрагмент тексту програми на мові C++. В кожному рядку тексту - один оператор. Замінити кожен оператор присвоювання виду "змінна = змінна + число;" на оператор скороченого присвоювання виду "змінна += число;". Визначити кількість змінених операторів і записати цю величину як значення «число» у останній змінений оператор присвоювання вхідного тексту. Вивести вміст вхідного і зміненого файлів.

main.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include "Functions2.h"
#include "Functions.h"

using namespace std;

int main(int argc, char * argv[])
{
    if(strcmp(argv[2], "FilePointer") == 0){
        FilePointerFunctions();
    }
    else if(strcmp(argv[2], "FileStream") == 0){
        FileStreamFunctions();
    }
    else{
        cout<<"Irregular expression"<< endl;
    }
}
```

Functions.cpp

```
#include <string>
#include <fstream>
#include <iostream>
#include "Functions.h"
#include "Functions2.h"

using namespace std;

string captureText(){
    cout << "Enter your text here :\nPress Enter to go to the next line\nPress
```

```

'-' + Enter on the new line to end writing\n";
string text;
string line;
// ascii код символу "-"
int asciiCode = 45;
// поки перший символ строки не "-"
while(int(line[0]) != asciiCode){
    getline(cin, line);
    text += line + '\n';
}
// відкидаємо 3 зайвих символи (2 зайвих \n і ^D)
text = text.substr(0, text.size()-3);

return text;
}

void replaceNumber(int counter){
    fstream fileRead("output.txt");
    string text;
    string line;

    while (getline(fileRead, line))
    {
        text += line + '\n';
    }
    fileRead.close();

    text = text.substr(0, text.size() - 1);
    int lastSpace = text.rfind(' ');
    int lastPositionSemicolon = text.rfind(';');
    int lengthTheLastNumber = lastPositionSemicolon - lastSpace - 1;

    text.replace(lastSpace + 1, lengthTheLastNumber, to_string(counter));

    ofstream fileWrite("output.txt");
    fileWrite << text;
    fileWrite.close();
}

void getOutputChangedFile(fstream &outputFile, string &stringLine) {
    cout << "\nOutput after changing file:\n";
    outputFile.open("output.txt");
    while (getline(outputFile, stringLine))
    {
        cout << stringLine << "\n";
    }
    outputFile.close();
}

void getChangedOperators(int count) {cout << "\nNumber of changed operators: " << count << "\n"; }

void getOutputFile() {
    ifstream inputFile("output.txt");
    if (!inputFile.is_open()) {
        cout << "Error opening file." << endl;
        return;
    }
}

```

```

    cout << "\nOutput before changing file:\n";

    string line;
    while (getline(inputFile, line)) {
        cout << line << endl;
    }

    inputFile.close();
}

string &getInputFile(ifstream &inputFile, string &stringLine) {
    cout << "Input file:\n";
    inputFile.open("input.txt");
    while (getline(inputFile, stringLine))
    {
        cout << stringLine << "\n";
    }
    inputFile.close();
    return stringLine;
}

void changeOperator(ifstream &inputFile, fstream &outputFile, string
&stringLine, int &count) {
    while (getline(inputFile, stringLine))
    {
        // перевіряємо, чи містить рядок оператор присвоєння виду "змінна =
змінна + число;"
        size_t pos = stringLine.find(" = ");

        if (pos != string::npos && stringLine.find('+') != string::npos)
//npos the biggest value of size_t
        {
            // замінюємо оператор на "змінна += число;"

            int firstElementPos = 0;
            int firstSpace = stringLine.find(' ');
            int quantityOfLetters = firstSpace - firstElementPos;
            stringLine.replace(pos, quantityOfLetters + 6, " += ");
            count++;

        }

        // записуємо рядок у вихідний файл
        outputFile << stringLine << "\n";

    }
}

void FileStreamFunctions() {
    string user_input = captureText();
    ofstream inputFileToWrite("input.txt");
    inputFileToWrite << user_input;
    inputFileToWrite.close();

    ifstream inputFile("input.txt");// for reading
    fstream outputFile("output.txt");//for writing

    string stringLine;
    int count = 0;

    changeOperator(inputFile, outputFile, stringLine, count);//кінець

```

```

        inputFile.close();
        outputFile.close();

        stringLine = getInputFile(inputFile, stringLine);

        getOutputFile();
        getChangedOperators(count);

        replaceNumber(count);

        getOutputChangedFile(outputFile, stringLine);
    }

void FilePointerFunctions() {
    string user_input = captureText2();
    FILE* inputFileToWrite = fopen("input.txt", "w");

    fprintf(inputFileToWrite, "%s\n", user_input.c_str());

    fclose(inputFileToWrite);

    FILE* inputFile = fopen("input.txt", "r");

    FILE* outputFile = fopen("output.txt", "w");

    string stringLine;
    int count = 0;

    changeOperator2(inputFile, outputFile, user_input, &count);

    fclose(inputFile);
    fclose(outputFile);

    getInputFile2("input.txt", stringLine);
    getOutputFile2((string &) "output.txt", stringLine);
    getChangedOperators2(count);
    replaceNumber2(count);
    getOutputChangedFile2("output.txt");
}

```

Functions.h

```

using namespace std;

#ifndef LABA_1_FUNCTIONS_H
#define LABA_1_FUNCTIONS_H

string captureText();

void replaceNumber(int counter);

void getOutputChangedFile(fstream &outputFile, string &stringLine);

void getChangedOperators(int count);

void getOutputFile();
string &getInputFile(ifstream &inputFile, string &stringLine);

void changeOperator(ifstream &inputFile, fstream &outputFile, string

```

```

&stringLine, int &count);
void FilePointerFunctions();
void FileStreamFunctions();
#endif //LABA_1_FUNCTIONS_H

```

Functions2.cpp

```

#include <iostream>
#include <cstring>
#include <string>
#include "Functions2.h"
using namespace std;
string captureText2(){
    cout << "Enter your text here :\nPress Enter to go to the next
line\nPress '-' + Enter on the new line to end writing\n";
    string text;
    string line;
    // ascii код символу "-"
    int asciiCode = 45;
    // поки перший символ строки не "-"
    while(int(line[0]) != asciiCode){
        getline(cin, line);
        text += line + '\n';
    }
    // відкидаємо 3 зайвих символи (2 зайвих \n і ^D)
    text = text.substr(0, text.size()-3);

    return text;
}

void replaceNumber2(int counter) {
    FILE* fileRead = fopen("output.txt", "r");
    if (fileRead == NULL) {
        std::cout << "Error opening file for reading." << std::endl;
        return;
    }
    fseek(fileRead, 0, SEEK_END);
    long fileSize = ftell(fileRead);
    rewind(fileRead);

    char* text = new char[fileSize + 1];
    std::memset(text, 0, fileSize + 1);

    fread(text, sizeof(char), fileSize, fileRead);

    fclose(fileRead);

    int lastSpace = string(text).rfind(' ');
    int lastPositionSemicolon = string(text).rfind(';');
    // int lengthTheLastNumber = lastPositionSemicolon - lastSpace - 1;

    string newText = string(text).substr(0, lastSpace + 1) +
std::to_string(counter) + std::string(text).substr(lastPositionSemicolon);
    delete[] text;

    FILE* fileWrite = fopen("output.txt", "w");
    if (fileWrite == NULL) {
        cout << "Error opening file for writing." << std::endl;
        return;
    }
}

```

```

        fwrite(newText.c_str(), sizeof(char), newText.size(), fileWrite);

        fclose(fileWrite);
    }

    void getOutputChangedFile2(const std::string& filename) {
        std::cout << "\nOutput after changing file:\n";

        FILE* file = fopen(filename.c_str(), "r");

        if (file == nullptr) {
            std::cout << "Error opening file." << std::endl;
            return;
        }

        char buffer[1024];
        while (fgets(buffer, 1024, file) != nullptr) {
            std::cout << buffer;
        }

        fclose(file);
    }

    void getChangedOperators2(int count) {cout << "\nNumber of changed
operators: " << count << "\n"; }

    void getOutputFile2(string &stringLine, string basicString) {
        FILE *outputFile;
        char line[1000];

        outputFile = fopen("output.txt", "r");

        if (outputFile == NULL) {
            printf("Error opening file.\n");
            return;
        }

        printf("\nOutput before changing file:\n");

        while (fgets(line, 1000, outputFile)) {
            printf("%s\n", line);
        }

        fclose(outputFile);
    }

    std::string& getInputFile2(const std::string& filename, std::string&
stringLine) {
        std::cout << "Input file:\n";

        FILE* file = fopen(filename.c_str(), "r");

        if (file == nullptr) {
            std::cout << "Error opening file." << std::endl;
            return stringLine;
        }

        char buffer[1024];

```

```

        while (fgets(buffer, 1024, file) != nullptr) {
            std::cout << buffer;
            stringLine += buffer;
        }

        fclose(file);
        return stringLine;
    }

void changeOperator2(FILE *inputFile, FILE *outputFile, string stringLine,
int *count) {
    char c;
    int i = 0;
    while ((c = fgetc(inputFile)) != EOF) {
        if (c == '\n') {
            stringLine[i] = '\0';
            i = 0;

            // перевіряємо, чи містить рядок оператор присвоєння виду
            "змінна = змінна + число;"
            size_t pos = stringLine.find(" = ");

            if (pos != string::npos && stringLine.find('+') !=
string::npos) //npos the biggest value of size_t
            {
                // замінюємо оператор на "змінна += число;"
                int firstElementPos = 0;
                int firstSpace = stringLine.find(' ');
                int quantityOfLetters = firstSpace - firstElementPos;
                stringLine.replace(pos, quantityOfLetters + 6, " += ");
                (*count)++;
            }

            // записуємо рядок у вихідний файл
            fprintf(outputFile, "%s\n", stringLine.c_str());
        } else {
            stringLine[i++] = c;
        }
    }
}

```

Functions2.h

```

#ifndef LABA_1_FUNCTIONS2_H
#define LABA_1_FUNCTIONS2_H
using namespace std;

string captureText2();
void replaceNumber2(int counter);
void getOutputChangedFile2(const std::string& filename);
void getChangedOperators2(int count);
void getOutputFile2(string &stringLine, string basicString);
std::string& getInputFile2(const std::string& filename, std::string&
stringLine);
void changeOperator2(FILE *inputFile, FILE *outputFile, string stringLine,
int *count);
#endif //LABA_1_FUNCTIONS2_H

```


РЕЗУЛЬТАТИ ТЕСТУВАННЯ

mode FilePointer

```
Last login: Wed Mar  8 11:25:52 on ttys000
[mac@MacBook-Pro-mac cmake-build-debug % ./Laba_1 -mode FilePointer
Enter your text here :
Press Enter to go to the next line
Press '-' + Enter on the new line to end writing
a = a + 8;
bbb = bbb + 9;
bcbc = bcbc + 999;
-
Input file:
a = a + 8;
bbb = bbb + 9;
bcbc = bcbc + 999;

Output before changing file:
a += 8;

bbb += 9;

bcbc += 999;

Number of changed operators: 3

Output after changing file:
a += 8;
bbb += 9;
bcbc += 3;
```

mode FileStream

```
[mac@MacBook-Pro-mac cmake-build-debug % ./Laba_1 -mode FileStream
Enter your text here :
Press Enter to go to the next line
Press '-' + Enter on the new line to end writing
ghgh = ghgh + 67;
nnn = nnn + 67890;
dfdf = dfdf + 78;
f = f + 7;
-
Input file:
ghgh = ghgh + 67;
nnn = nnn + 67890;
dfdf = dfdf + 78;
f = f + 7;

Output before changing file:
ghgh += 67;
nnn += 67890;
dfdf += 78;
f += 7;

Number of changed operators: 4

Output after changing file:
ghgh += 67;
nnn += 67890;
dfdf += 78;
f += 4;
```

Незрозумілий ввід

```
[mac@MacBook-Pro-mac cmake-build-debug % ./Laba_1 -mode FileType  
Irregular expression  
mac@MacBook-Pro-mac cmake-build-debug % █
```

Лінк на репозиторій у GitHub:

<https://github.com/Uliana200407/CppProjects-.git>