

Отчёт к лабораторной работе

Подготовила Маркова Варвара Владимировна СКБ221

Москва 2022

Файлы

Файл *main.cpp*: функция `main` с аргументами `int argc`, `char** argv[]`.

Если (`argc == 1`), то выводится ошибка (нет флагов).

Если (`argc == 2`), то выводится ошибка (недостаточно аргументов).

Если (`argc == 3` и `argv[1] == "--tofile"`), то происходит ввод в консоль, а затем запись в файл, название которого содержится в `argv[2]` (вызывается функция `readfromcin`, `readtofile` и функция `task`, выполняющая само задание (её описание чуть ниже)).

Если (`argc == 3` и `argv[1] == "--fromfile"`), то происходит считывание данных из файла, название которого содержится в `argv[2]` (вызывается функция `readfromfile` и функция `task`).

Если (`argc == 4`, и `argv[1] == "--tofile"`, и `argv[3] == "--fromfile"`), то происходит ввод данных в консоль, затем запись в файл, название которого содержится в `argv[2]`, а затем считывание из этого же файла (вызывается функция `readtofile`, `readfromfile` и функция `task`).

Если (`argc == 5`, и `argv[1] == "--fromfile"`, и `argv[3] == "--tofile"`), то происходит считывание данных из файла, название которого содержится в `argv[2]`, а затем запись считанных данных в файл, название которого содержится в `argv[4]` (вызывается функция `readtofile`, `readfromfile` и функция `task`).

В остальных случаях выводится ошибка (неверные флаги).

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include "functions.h"
using std::cout;
using std::endl;
using std::cerr;
const char* flag[2] = { "--tofile", "--fromfile" };
int main(int argc, char** argv) {
    if (argc == 1) {
        cerr << "Error: there are no flags" << endl;
    }
    else if (argc == 2) {
        cerr << "Error: not much segments" << endl;
    }
    else if (argc == 3) {
        char* name = argv[2];
        if (!(strcmp(argv[1], flag[0]))) {
            readfromcin();
            readtofile(name);
        }
    }
}
```

```

        task();
    }
    else if (!strcmp(argv[1], flag[1])) {
        readfromfile(name);
        task();
    }
    else cerr << "Wrong flags" << endl;
}
else if (argc == 4) {
    if (!strcmp(argv[1], flag[0])) {
        if (!strcmp(argv[3], flag[1])) {
            char* name = argv[2];
            readfromcin();
            readtofile(name);
            readfromfile(name);
            task();
        }
        else {
            cerr << "Wrong flags" << endl;
        }
    }
    else {
        cerr << "Wrong flags" << endl;
    }
}
else if (argc == 5) {
    if ((!strcmp(argv[1], flag[1]))) {
        if ((!strcmp(argv[3], flag[0]))) {
            char* name1 = argv[2];
            char* name2 = argv[4];
            readfromfile(name1);
            readtofile(name2);
            task();
        }
        else {
            cerr << "Wrong flags" << endl;
        }
    }
    else {
        cerr << "Wrong flags" << endl;
    }
}
else {
    cerr << "Wrong flags" << endl;
}
}

```

Файл *functions.cpp*: задаётся массив `vyr` типа `char`, в котором будут храниться все символы данных. Функции `readfromcin` считывает с консоли введённые данные, `readtofile` записывает в файл данные, заменяя перенос строки на пробел, `readfromfile` считывает из файла данные, заменяя перенос строки на пробел, `bubblesort` осуществляет пузырьковую сортировку. Функция `task` осуществляет условие задания: проходит по символьному массиву, вычисляя длину каждого предложения и заносит это значение в массив `arg`, который впоследствии сортируется по возрастанию с помощью пузырьковой сортировки. А массив `mas` представляет собой изначальный массив `arg`, ещё не подвергшийся сортировке. Массив `sum` имеет следующий вид: первый его элемент – это первый элемент `mas`, а следующие его элементы – это сумма предыдущего элемента `mas` и настоящего (массив `sum` создаётся для обработки индексов элементов, обозначающих концы предложений). Благодаря отсортированному массиву `arg` мы понимаем, в каком порядке выводить предложения по возрастанию их длин. Также в функции `task` реализуется запись в файл `NINE.txt` самого длинного предложения в обратном порядке (делается это тоже с помощью прохода по элементам символьного массива).

```
#include <iostream>
#include <cstring>
#include <fstream>

using std::cout;
using std::endl;
using std::cin;
using std::ifstream;
using std::ofstream;

const int len = 600;
char vyr[len];

void readfromcin() {
    cin.read(vyr, len - 1);
    int s = cin.gcount();
    vyr[s] = '\0';
}

void bubblesort(int* num, int raz) {
    for (int i = 0; i < raz - 1; i++) {
        for (int j = (raz - 1); j > i; j--) {
            if (num[j - 1] > num[j]) {
                int temp = num[j - 1];
                num[j - 1] = num[j];
                num[j] = temp;
            }
        }
    }
}
```

```

    }
}

void readtofile(char* name) {
    ofstream par;
    par.open(name);
    if (par.is_open()) {
        int i = 0;
        while (vyr[i]) {
            if (vyr[i] == '\n') vyr[i] = ' ';
            par << vyr[i];
            i++;
        }
    }
    else { cout << "No open" << endl; }
    par.close();
}

void readfromfile(char* name) {
    int i = 0;
    ifstream file_in(name);
    noskipws(file_in);
    while (file_in >> vyr[i]) {
        if (vyr[i] == '\n') {
            vyr[i] = ' ';
            vyr[i - 1] = ' ';
        }
        ++i;
        if (file_in.peek() == EOF) {
            break;
        }
    }
    //j++;
    //}
    //int z = 0;
    //while (vyr[z]) {
        //cout << vyr[z] << endl;
        //z++;
    //}
}

void task() {
    int g = 0;
    int size = 0;
    while (vyr[g]) {
        size++;
        g++;
    }
}

```



```

    }
    p++;
    u = 0;
}
bubblesort(rt, m);
//for (int i = 0; i < m; i++) {
    //cout << rt[i] << endl;
//}
int* sum = new int[m];
for (int i = 0; i < m; i++) {
    if (i == 0) {
        sum[i] = mas[i];
    }
    else {
        sum[i] = sum[i - 1] + mas[i];
    }
}
//for (int i = 0; i < m; i++) {
    //cout << sum[i] << endl;
//}
for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
        if (rt[i] == mas[j]) {
            if (j != 0) {
                //if (i != 0) {
                int k = 0;
                int v = sum[j - 1] + 2 * j;
                if (vyr[v] == ' ') k = v + 1;
                else k = v;
                for (int l = k; l <= sum[j] + 2 * j; l++) {
                    if (vyr[l] == '\n') {
                        cout << ' ';
                    }
                    else {
                        if (l > k && vyr[l] == vyr[l - 1] && vyr[l] == ' ')
{continue;}

                        else {cout << vyr[l];}
                    }
                }
                cout << endl;
            }
            //else {
                //for (int k = sum[j - 1] + j + 1; k <= sum[j] + j; k++) {
                    //cout << str[k];
                //}
                //cout << endl;
            //}
//}
else {
    //if (i != 0) {
        for (int l = 0; l <= sum[j]; l++) {

```

```

        if (vyr[l] == '\n') {
            cout << ' ';
        }
        else {
            if (l > 0 && vyr[l] == vyr[l - 1] && vyr[l] == ' ')
{continue;}

            else {cout << vyr[l];}
        }
    }
    cout << endl;
}
//else {
    //for (int k = 0; k <= sum[j]; k++) {
        //cout << str[k];
    //}
    //cout << endl;
}
//}
}
}
int max = 0;
int f = 0;
for (int i = 0; i < m; i++) {
    if (mas[i] > max) {
        max = mas[i];
        f = i;
    }
}
//cout << max << endl;
//cout << u << endl;
ofstream out;
out.open("NINE.txt");
if (out.is_open()) {
    int w = 0;
    if (f != 0) {
        for (int i = sum[f] + 2 * f - 1; i >= sum[f - 1] + 2 * f; i--) {
            if (i != (sum[f - 1] + 2 * f)) {
                if (vyr[i] == ' ' || vyr[i] == '\n') {
                    for (int j = i + 1; j < i + w + 1; j++) {
                        out << vyr[j];
                    }
                    out << ' ';
                    w = 0;
                }
            }
            else {
                w++;
            }
        }
    }
    else {
        for (int j = sum[f - 1] + 2 * f; j < i + w + 1; j++) {
            out << vyr[j];
        }
    }
}

```



```

        }
        out << '.';
    }
}
}
else {
    for (int i = sum[f] - 1; i >= 0; i--) {
        if (i != (sum[f - 1] + 2 * f)) {
            if (vyr[i] == ' ' || vyr[i] == '\n') {
                for (int j = i + 1; j < i + w + 1; j++) {
                    out << vyr[j];
                }
                out << ' ';
                w = 0;
            }
            else {
                w++;
            }
        }
        else {
            for (int j = sum[f - 1] + 2 * f; j < i + w + 1; j++) {
                out << vyr[j];
            }
            out << '.';
        }
    }
}
}
else { cout << "No open" << endl; }
out.close();

delete[] arr;
delete[] mas;
delete[] sum;
delete[] rt;
}

```

Файл *functions.h*: в этом заголовочном файле определены все используемые функции.

```

#ifndef FUNCTIONS
#define FUNCTIONS
void readfromcin();
void readtofile(char*);
void readfromfile(char*);
void task();
#endif

```

Файл Makefile: реализация сборки, clean и distclean.

```
OBJS=functions.o main.o

prog: $(OBJS)
    g++ -o prog $(OBJS)

.PHONY: clean
clean:
    $(RM) $(OBJS)

distclean: clean
    $(RM) prog
```

Файл output.txt: содержит пример из условия задания.

Alice's Adventures in Wonderland is a children's novel. The title is usually shortened to Alice in Wonderland. Charles Lutwidge Dodgson wrote the book. He wrote it using the pen name Lewis Carroll. John Tenniel drew the 42 pictures in the book. The book was published by Macmillan and Co in London. It was released on 26 November 1865. It has been adapted to numerous movies. In 2010, it has been adapted to Hollywood movie.

Файл NINE.txt: записывается самое длинное предложение в обратном порядке (для примера из условия: novel children's a is Wonderland in Adventures Alice's.)

Файл qwerty.txt: содержит пример с наличием переноса строки в файле (который в ходе выполнения программы заменится на пробел).

Hello

everyone. Ab. Cde.

Вывод

Я продолжила учиться работать с командной строкой и с make.



