**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №5**

**по дисциплине «Программирование»**

Тема: Файлы для ввода-вывода данных, двумерные массивы. Функции

|  |  |  |
| --- | --- | --- |
| Студент гр. 4353 |  | Аксёнов Д.А. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2024

Оглавление:

1. Формулировка задания 3

2. Устранение неточностей 3

3. Математическая постановка задачи 3

4. Контрольный пример 3

5. Ограничения 3

6. Разработка интерфейса пользователя 3

7. Библиотеки 4

8. Разработка интерфейса пользователя 4

9. Алгоритм 5

10. Текст программы 9

11. Результаты работы программы 11

12. Вывод 11

1. **Формулировка задания**

Составить программу, которая считывает данные из файла, составляет из этих данных двумерный массив А целых чисел размера n x m и переставляет в нём элементы так, чтобы сначала следовали чётные элементы, а затем нечётные.

1. **Устранение неточностей**

Массив n x m составляется из элементов файла, где лишние элементы отсекаются.

1. **Математическая постановка задачи**

*Дано:* файл с некоторыми числами.

*Преобразовать:* сначала следуют все чётные элементы, а затем нечётные.

*Способ решения:* для начала создадим временные массивы для чётных и нечётных элементов. Пусть i – номер строки, j – номер столбца, тогда можно заполнить их с помощью цикла i < n с единичным шагом и вложенного цикла j < n с единичным шагом: для чётных элементов выполняется условие (static\_cast<int>(A[i][j]) % 2 == 0), а для нечётных – (static\_cast<int>(A[i][j]) % 2 != 0). Переписываем элементы обратно в матрицу (сначала чётные, потом нечётные) с помощью циклов i < evenIndex (A[index / b][index % b] = evenElements[i]) и i < oddIndex (A[index / b][index % b] = oddElements[i]) с единичным шагом, при каждом шаге увеличивая значение index.

1. **Контрольный пример**

Файл in.txt:

21 145 25

662 0

215 67

41 67 64 5 sss

999 113 56

king

12 268

Начальный массив:

21 145

662 0

215 67

41 560

999 112

Итоговый массив:

662 0

560 112

21 145

215 67

41 999

1. **Ограничения**

Ограничения на ввод [10-20;1020].

1. **Разработка интерфейса пользователя**

O1: Автор: Аксёнов Даниил Андреевич, Группа: 4353

Версия: 1, Дата: начало 14.11.2024, завершение - 14.11.2024

Задание: 11

O2: Строки a: ±ddd; столбцы b: ±ddd

O3: Начальный массив: ±ddd.ddd ±ddd.ddd …

Итоговый массив: ±ddd.ddd ±ddd.ddd …

O4: Массив пустой

1. **Библиотеки**

|  |  |  |
| --- | --- | --- |
| Библиотека | Функция | Назначение |
| iostream | cout << | Вывод текста |
| fstream | open() | Открыть файл |
| close() | Закрыть файл |
| clear() | Очищает ввод ошибок |
| eof() | Возвращает истину, если это конец файла, иначе ложь |
| iomanip | resetiosflags() | Удаляет указанные флаги |

1. **Внутреннее представление данных в программе**

Переменные:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Назначение |
| i | int | Индекс строки массива |
| j | Индекс столбца массива |
| a | unsigned | Количество строк массива |
| b | Количество столбцов массива |
| h | Промежуточное количество строк |
| w | Промежуточное количество столбцов |
| i | Индекс строки массива |
| j | Индекс столбца массива |
| evenIndex | Индекс чётного элемента |
| oddIndex | Индекс нечётного элемента |
| index | Индекс элемента в массиве |
| A | float | Массив |
| t | Текущий элемент массива |
| evenElements | Временный массив для чётных элементов |
| oddElements | Временный массив для нечётных элементов |
| s | char | Текущий символ |
| name | Текст файла |
| f | fstream | Файлы на ввод и вывод |
| N | Const unsigned | Максимальное количество строк массива |
| M | Максимальное количество столбцов массива |

Функции:

* Синтаксис

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип возвращаемого значения | Параметры | | |
| Тип | Имя | Изменение |
| size | void | char | name | - |
| unsigned | a | & |
| unsigned | b | & |
| inp | void | char | name | - |
| float | A | - |
| unsigned | a | - |
| unsigned | b | - |
| process | void | float | A | - |
| unsigned | a | - |
| unsigned | b | - |
| out | void | float | A | - |
| unsigned | a | - |
| unsigned | b | - |
| fstream | f | & |
| outline | void | float | A | - |
| unsigned | b | - |
| fstream | f | & |

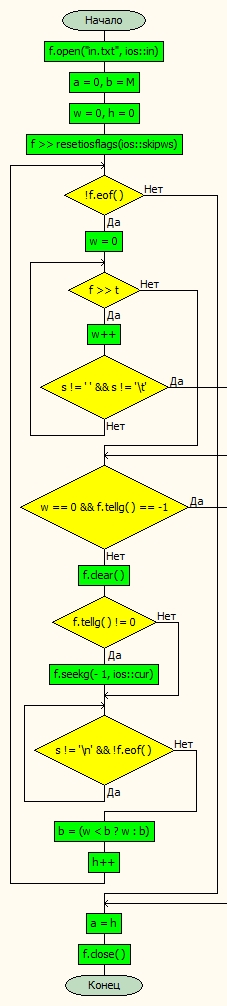
* Назначения

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | Возвращаемое значение | Параметры | | | | Внешние изменения |
| Входной | Выходной | Модифицируемый | Транзитный |
| process |  | A, a, b |  |  |  |  |
| size |  | name |  | a, b |  |  |
| inp |  | A, name, a, b | A |  |  |  |
| out |  | A, a, b |  | f |  |  |
| outline |  | A, b |  | f |  |  |

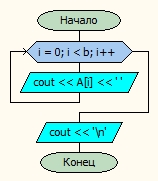
1. **Алгоритм**

В программе сначала находится размер массива функцией size, далее если размер равен 0, то программа завершается принудительно. Если размер массива равен не 0, то массив преобразуется. Далее считываются элементы из файла в массив функцией inp, далее выводится начальный массив, затем он преобразуется и выводится снова в файл out.txt и в консоль функцией out.

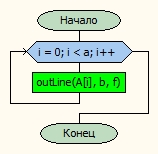
size:



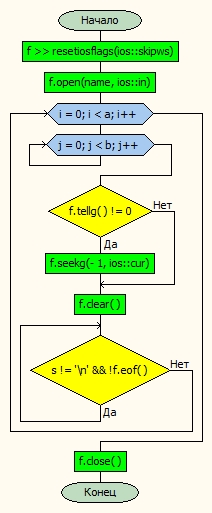
outLine:



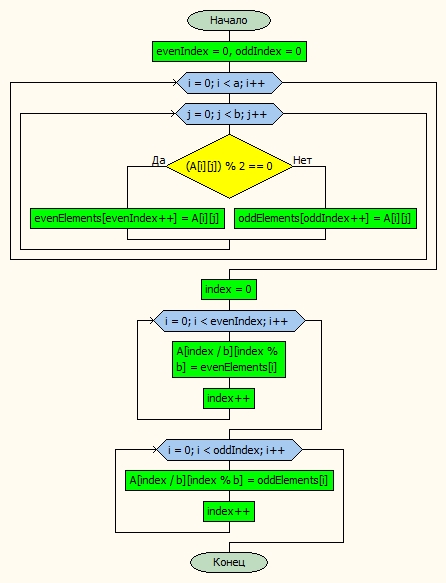
out:



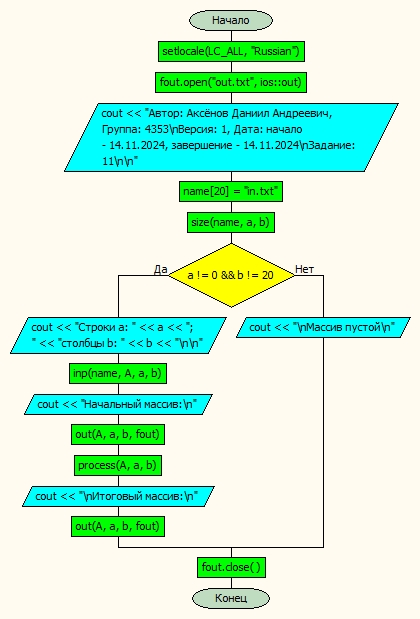
inp:



process:



main:



1. **Текст программы**

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

const unsigned N = 30, M = 20;

void size(char name[20], unsigned& a, unsigned& b) {

fstream f;

f.open("in.txt", ios::in);

char s;

a = 0, b = M;

unsigned w = 0, h = 0;

float t;

f >> resetiosflags(ios::skipws);

while (!f.eof()) {

w = 0;

while (f >> t) {

f >> s;

w++;

if (s != ' ' && s != '\t') break;

}

if (w == 0 && f.tellg() == -1) break;

f.clear();

if (f.tellg() != 0) f.seekg(-1, ios::cur);

f >> s;

while (s != '\n' && !f.eof()) f >> s;

b = (w < b ? w : b);

h++;

}

a = h;

f.close();

}

void outLine(float A[M], unsigned b, fstream& f) {

for (int i = 0; i < b; i++) {

cout << A[i] << ' ';

f << A[i] << ' ';

}

cout << '\n';

f << '\n';

}

void out(float A[N][M], unsigned a, unsigned b, fstream& f) {

for (int i = 0; i < a; i++) outLine(A[i], b, f);

}

void inp(char name[20], float A[N][M], unsigned a, unsigned b) {

fstream f;

char s;

f >> resetiosflags(ios::skipws);

f.open(name, ios::in);

for (int i = 0; i < a; i++) {

for (int j = 0; j < b; j++) {

f >> A[i][j];

f >> s;

}

if (f.tellg() != 0) f.seekg(-1, ios::cur);

f >> s;

f.clear();

while (s != '\n' && !f.eof()) f >> s;

}

f.close();

}

void process(float A[N][M], unsigned a, unsigned b) {

float evenElements[N \* M];

float oddElements[N \* M];

unsigned evenIndex = 0, oddIndex = 0;

for (unsigned i = 0; i < a; i++) {

for (unsigned j = 0; j < b; j++) {

if (static\_cast<int>(A[i][j]) % 2 == 0) {

evenElements[evenIndex++] = A[i][j];

}

else {

oddElements[oddIndex++] = A[i][j];

}

}

}

unsigned index = 0;

for (unsigned i = 0; i < evenIndex; i++) {

A[index / b][index % b] = evenElements[i];

index++;

}

for (unsigned i = 0; i < oddIndex; i++) {

A[index / b][index % b] = oddElements[i];

index++;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

fstream fout;

fout.open("out.txt", ios::out);

cout << "Автор: Аксёнов Даниил Андреевич, Группа: 4353\nВерсия: 1, Дата: начало - 14.11.2024, завершение - 14.11.2024\nЗадание: 11\n\n";

fout << "Автор: Аксёнов Даниил Андреевич, Группа: 4353\nВерсия: 1, Дата: начало - 14.11.2024, завершение - 14.11.2024\nЗадание: 11\n\n";

float A[N][M];

unsigned a, b;

char name[20] = "in.txt";

size(name, a, b);

if (a != 0 && b != 20) {

cout << "Строки a: " << a << "; " << "столбцы b: " << b << "\n\n";

fout << "Строки a: " << a << "; " << "столбцы b: " << b << "\n\n";

inp(name, A, a, b);

cout << "Начальный массив:\n";

fout << "Начальный массив:\n";

out(A, a, b, fout);

process(A, a, b);

cout << "\nИтоговый массив:\n";

fout << "\nИтоговый массив:\n";

out(A, a, b, fout);

}

else {

cout << "\nМассив пустой\n";

fout << "\nМассив пустой\n";

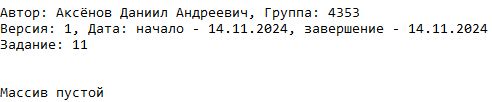
}

fout.close();

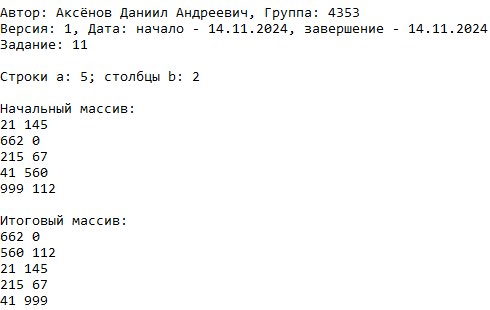
}

1. **Результаты работы программы**

Пустой массив:



Заполненный массив:



1. **Вывод**

В ходе выполнения задания были освоены функции и работа с двумерными массивами.