**Лабораторная работа №8**

**Бинарные файлы**

17 вариант

1. Создать файл, содержащий вещественные числа. Подсчитать количество элементов, меньших заданного пользователем значения.
2. Создать файл f, содержащий двумерный массив (3х3) целых чисел. Записать в файл g все отрицательные числа из этого массива.

**Текст программы**

1 задание:

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

float a;

printf("Текст файла:\n");

FILE\* f;

f = fopen("input.txt", "w+b");

for (int i = 1; i <= 10; i++) {

a = rand() / 100.0 - 150;

fwrite(&a, sizeof(a), 1, f);

}

fclose(f);

f = fopen("input.txt", "r+b");

for (int i = 1; i <= 10; i++) {

fread(&a, sizeof(a), 1, f);

printf("%f\n", a);

}

fclose(f);

float b; int c = 0;

printf("Введите число: ");

scanf("%f", &b);

f = fopen("input.txt", "r+b");

for (int i = 1; i <= 10; i++) {

fread(&a, sizeof(a), 1, f);

if (a<b) c++;

}

fclose(f);

printf("Количество элементов, меньших %f: %i\n", b, c);

system("pause");

return 0;

}

2 задание:

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

int a[3][3], n, k = 0;

ofstream f1;

f1.open("f.txt", ios::binary | ios::out);

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++) {

n = rand() % 100 - 50;

f1.write((char\*)&n, sizeof(n));

}

f1.close();

ifstream f2;

f2.open("f.txt", ios::binary | ios::in);

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

f2.read((char\*)&a[i][j], sizeof(a));

cout << a[i][j] << ' ';

}

cout << endl;

}

f2.close();

ofstream g1;

g1.open("g.txt", ios::binary | ios::out);

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

if (a[i][j] < 0) {

g1.write((char\*)&a[i][j], sizeof(n));

k++;

}

g1.close();

ifstream g2;

g2.open("g.txt", ios::binary | ios::in);

if (k == 0) cout << "Отрицательных элементов нет" << endl;

else {

cout << "Отрицательные элементы массива:" << endl;

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

g2.read((char\*)&n, sizeof(n));

cout << n << endl;

}

}

g2.close();

system("pause");

return 0;

}

**Описание программы**

1 задание:

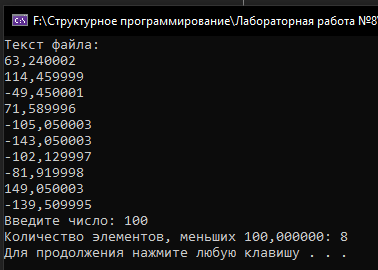
1. Вывод надписи «Текст файла:»
2. Описывается переменная f типа FILE\*
3. Открывается файл input.txt (если его нет – он создается, если есть - заменяется), режим w+b (открытие бинарного файла для записи)
4. Задается цикл for, в котором переменная i принимает значения от 1 до 10
5. Переменная a принимает рандомное значение
6. Значение переменной a записывается в файл
7. Файл input.txt закрывается
8. Открывается файл input.txt, режим r+b (открытие бинарного файла для чтения)
9. Задается цикл for, в котором переменная i принимает значения от 1 до 10
10. Переменной a присваивается значение из файла
11. Значение переменной a выводится на экран
12. Файл input.txt закрывается
13. Задаются переменные b и c, переменной c присваивается значение 0
14. Программа просит пользователя ввести число b
15. Открывается файл input.txt, режим r+b (открытие бинарного файла для чтения)
16. Задается цикл for, в котором переменная i принимает значения от 1 до 10
17. Переменной a присваивается значение из файла
18. Если a<b, то значение переменной c увеличивается на 1
19. Файл input.txt закрывается
20. Выводится значение переменной c

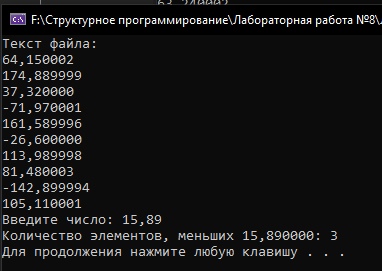
2 задание:

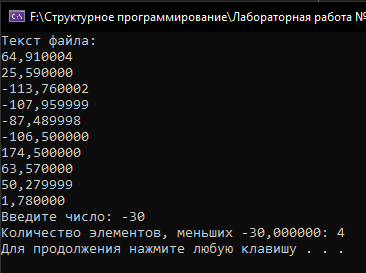
1. Задается двумерный статический массив a размером 3х3
2. Задаются переменные n и k, переменной k присваивается значение 0
3. Описывается поток f1 типа ofstream
4. Открывается файл f.txt (если его нет – он создается, если есть - заменяется), режимы binary, out (открытие бинарного файла для записи)
5. Задается цикл for, в котором переменная i принимает значения от 0 до 2
6. В цикле for задается еще один цикл for, в котором переменная j принимает значения от 0 до 2
7. Переменной n присваивается рандомное значение
8. Значение переменной n записывается в файл
9. Файл f.txt закрывается
10. Описывается поток f2 типа ifstream
11. Открывается файл f.txt, режимы binary, in (открытие бинарного файла для чтения)
12. Задается цикл for, в котором переменная i принимает значения от 0 до 2
13. В цикле for задается еще один цикл for, в котором переменная j принимает значения от 0 до 2
14. Элементу массива a[i][j] присваивается значение из файла
15. Элемент массива a[i][j] выводится на экран
16. Файл f.txt закрывается
17. Описывается поток g1 типа ofstream
18. Открывается файл g.txt (если его нет – он создается, если есть - заменяется), режимы binary, out (открытие бинарного файла для записи)
19. Задается цикл for, в котором переменная i принимает значения от 0 до 2
20. В цикле for задается еще один цикл for, в котором переменная j принимает значения от 0 до 2
21. Если a[i][j]<0, то значение этого элемента записывается в файл
22. Значение переменной k увеличивается на 1
23. Файл g.txt закрывается
24. Описывается поток g2 типа ifstream
25. Открывается файл g.txt, режимы binary, in (открытие бинарного файла для чтения)
26. Если k=0, выводится надпись «Отрицательных элементов нет» и далее шаг 31, иначе шаг 27
27. Выводится надпись «Отрицательные элементы массива:»
28. Задается цикл for, в котором переменная i принимает значения от 0 до k
29. Переменной n присваивается значение из файла
30. Значение переменной n выводится на экран
31. Файл g.txt закрывается

**Результаты тестов**

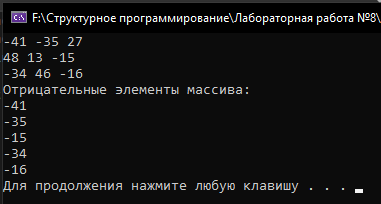
1 задание:

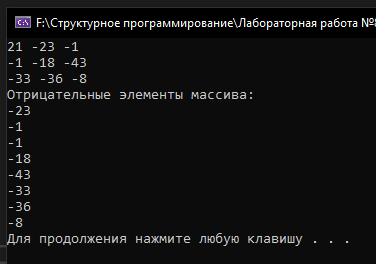


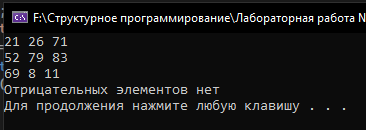




2 задание:







**Вывод**

Мною были написаны 2 программы, которые формируют бинарные файлы, записывают в них информацию, и затем ее читают.

Первая программа написана средствами языка C, а вторая – средствами языка C++. Алгоритм программ идентичен.

Первая программа создает файл, и заполняет его вещественными числами, которые затем выводит на экран. Далее программа подсчитывает количество элементов, меньших заданного пользователем значения

Вторая программа создает файл, в который записывается двумерный массив. Этот массив выводится на экран. Затем в другой файл записываются все отрицательные элементы массива, а потом они выводятся на экран.