

Министерство высшего образования и науки РФ
ФГБОУ ВО ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Информационная безопасность систем и технологий»

ОТЧЁТ

о лабораторной работе №5

Работа с файлами в стандартной библиотеке Си++

Дисциплина: Языки программирования

Группа: 18ПИ1

Выполнил: Асаян А.В.

Количество баллов:

Дата сдачи:

Проверил: к.т.н., доцент Лупанов М.Ю.

Пенза, 2019

1 Цель работы

1.1 Освоить операции форматированного и блочного ввода-вывода с помощью библиотеки языка Си++.

2 Задания к практической работе

2.1 Написать программу, читающую последовательность из десяти целых чисел, а затем выводящих их в виде трех столбцов, в первом столбце должны быть выведены восьмеричные значения этих чисел, во втором — десятичные, в третьем — шестнадцатеричные.

2.2 Написать функцию для вывода таблицы умножения заданной системы счисления. Систему счисления задавать в качестве параметра функции. Обеспечить поддержку восьмеричной, десятичной и шестнадцатеричной систем.

2.3 Написать программу, читающую числа с плавающей точкой из заданного преподавателем тестового файла, и вычисляющую их среднее арифметическое.

2.4 Написать программу, читающую содержимое одного текстового файла, и записывающую его в конец другого текстового файла. Файлы для работы создать самостоятельно.

2.5 Написать программу, читающую из заданного преподавателем двоичного файла массив чисел типа double, формирующую массив чисел типа float с обратными значениями ($f(x)=1/x$) и записывающую полученный массив в новый двоичный файл.

3 Результат выполнения работы

3.1 Была написана программа, считывающая последовательность из десяти целых чисел и выводящая их на экран в три столбца. В программе сначала открывается файл из которого читает в массив последовательность из 10 чисел, после чего с помощью манипуляторов вывода на экран выводятся элементы массива в разных системах счисления. На рисунке 1 представлен результат работы программы. Код программы:

```

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int main(int argc, char **argv)
{
    ifstream      fit("posled.txt");//,ios::out
ios::binary);
    //string b;
    int c[10];
    if(!fit.is_open()){
        cerr << "Error: file is not open\n";
        return 1;
    }
    for(int i=0;i<10;i++)
        fit>>c[i];
    for(int i=0;i<10;i++){
        cout<<setw(10)<<oct<<c[i]<<setw(10)<<dec<<c[i]
]<<setw(10)<<hex<<c[i]<<endl;

    }
    fit.close();
    return 0;
}

```

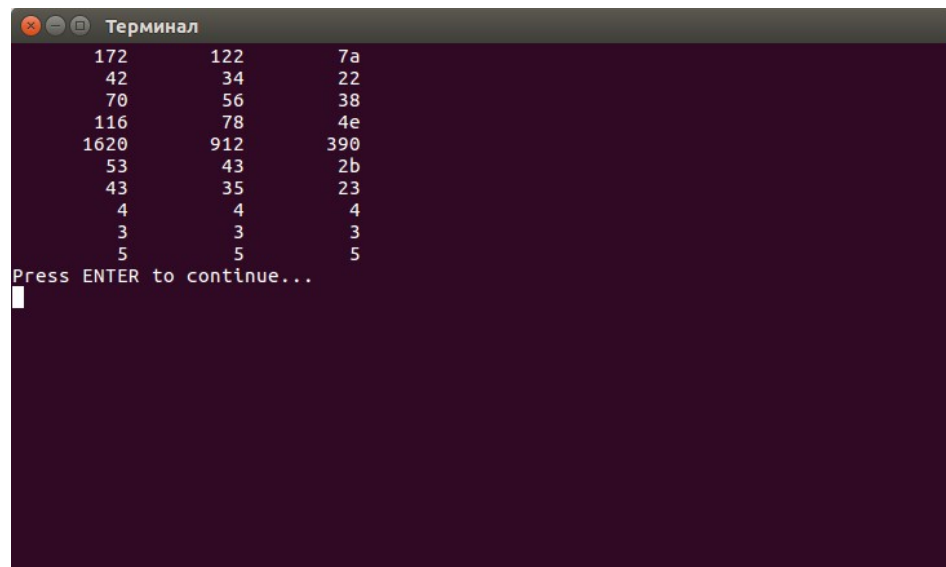


Рисунок 1 — Результат работы программы 1.

3.2 Была написана функция, выводящая на экран таблицу умножения в указанной системе счисления. На рисунке 2 представлен результат работы программы для шестнадцатеричной системы счисления. Код функции и программы, демонстрирующей её работу:

```

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
void tabl(int sischis)
{
    if(sischis==16) {
        for(int i=1; i<16; i++) {
            cout<<endl;
            for(int j=1; j<16; j++) {
                int z=i*j;
                cout<<setw(10)<<hex<<i<<'*'<<j<<'='<<
z<<endl;
            }
        }
    }
    if(sischis==8) {
        for(int i=1; i<8; i++) {
            cout<<endl;
            for(int j=1; j<8; j++) {
                int z=i*j;
                cout<<setw(10)<<oct<<i<<'*'<<j<<'='<<
z<<endl;
            }
        }
    }
    if(sischis==10) {
        for(int i=1; i<10; i++) {
            cout<<endl;
            for(int j=1; j<10; j++) {
                int z=i*j;
                cout<<setw(10)<<i<<'*'<<j<<'='<<z<<en
dl;
            }
        }
    }
}
int main(int argc, char **argv)
{
    int x;
    cin>>x;
    tabl(x);
    return 0;
}

```

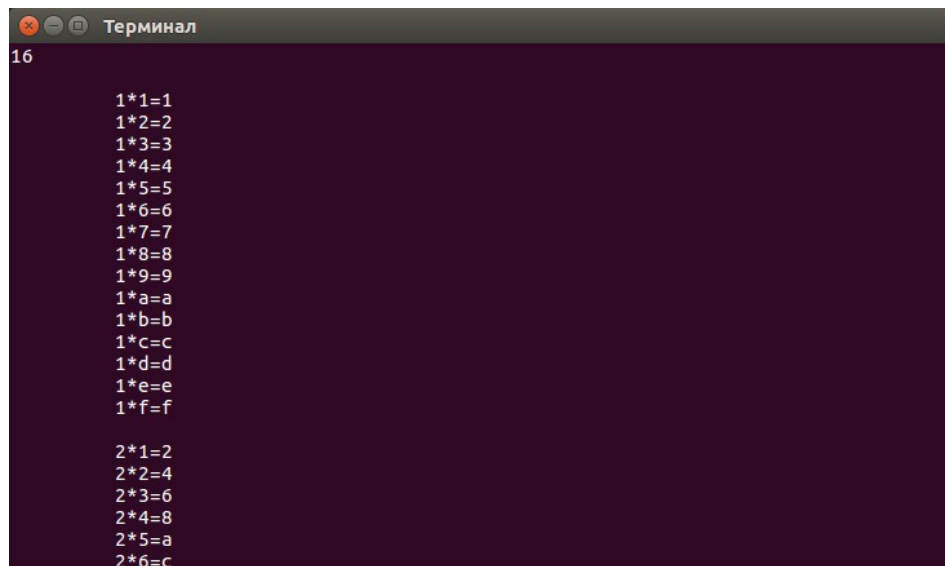


Рисунок 2 — Таблица умножения.

3.3 Была создана программа, которая читает из заданного файла числа с плавающей точкой в контейнер типа `vector`, после чего считает среднеарифметическое этих чисел. На рисунке три представлен результат работы программы. Код программы:

```

#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
int main(int argc, char **argv)
{
    ifstream fit("data_v5.txt");
    double number;
    int size=0;
    vector<double> chisla;
    if(!fit.is_open()){
        cerr << "Error: file is not open\n";
        return 1;
    }
    double sr=0;
    double sum=0;
    while(fit>>number)
    {
        chisla.push_back(number);
        size++;
    }

    for(int i=0;i<chisla.size();i++){

```

```

        sum=sum+chisla[i];
    }
    sr=sum/size;
    cout<<sr<<endl;
    fit.close();
    return 0;
}

```

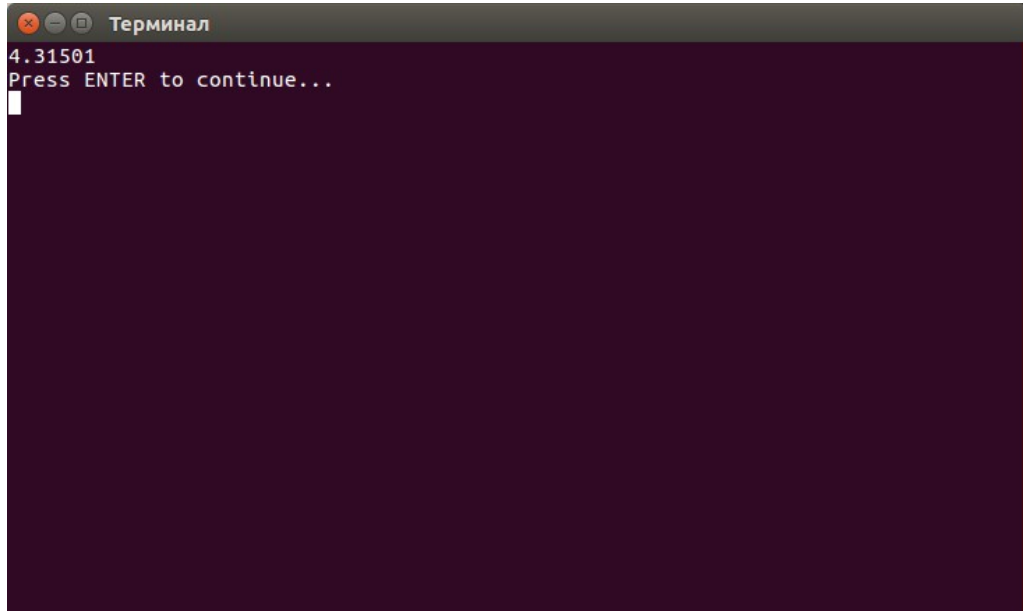


Рисунок 3 — Результат расчёта среднего арифметического.

3.4 Была написана программа, читающая содержимое одного текстового файла и записывающая его в конец другого. Программа открывает файл, в который будет производится запись и устанавливает позицию записи на конец файла. После открывается файл из которого читаются данные, позиция чтения переносится в конец файла, её значение присваивается переменной, содержащей количество символов в файле, после позиция снова возвращается в начало. Создаётся динамический массив, в который читаются символы из второго файла, после чего этот массив записывается в конец первого файла. Содержание исходных файлов представлено на рисунках 4 и 5. Результат работы программы представлен на рисунке 6. Код программы:

```

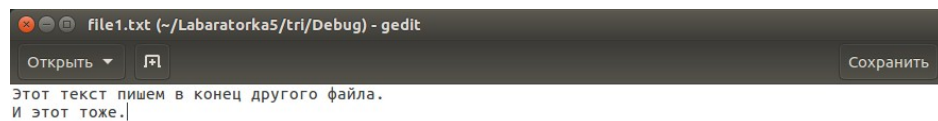
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int main(int argc, char **argv)

```

```

{
    fstream f("file2.txt",ios::in | ios::out);
    if(!f.is_open()) {
        cerr << "Error: file is not open\n";
        return 1;
    }
    f.seekp(0,ios::end);
    ifstream f1("file1.txt");
    if(!f1.is_open()) {
        cerr << "Error: file is not open\n";
        return 1;
    }
    f1.seekg(0,ios::end);
    int f1size = f1.tellg();
    //cout<<f1size;
    f1.seekg(0,ios::beg);
    char* buf = new char[f1size];
    f1.read(buf,f1size);
    f.write(buf,f1size);
    delete[]buf;
    return 0;
}

```



Текст Ширина табуляции: 8 Стр 2, Стлб 13 ВСТ

Рисунок 4 — Содержимое файла из которого производится копирование.

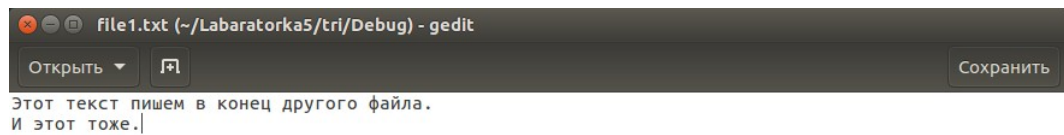


Рисунок 5 — Содержимое файла, в который производится запись.

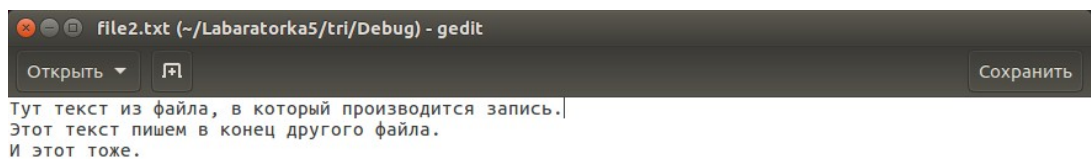


Рисунок 6 — Результат работы программы записи в файл.

3.5 Была написана программа, читающая в массив из двоичного файла числа типа double, формирующая массив типа float из обратных значений и записывающая их в другой двоичный файл. Сначала программа в двоичном виде открывает файл, в котором находятся числа, начинается подсчёт количества чисел в файле, который идёт до тех пор, пока возможно прочитать в переменную значение. После завершения подсчёта сбрасывается флаг и позиция чтения возвращается в начало файла. После чего создаётся массив с размером, равным количеству чисел в двоичном файле. Значения из файла записываются в массив. Создаётся ещё один массив типа float, в который записываются значения, обратные значениям первого массива. Эти значения записываются в новый двоичный файл. Код программы:

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int main(int argc, char **argv)
{
    double e=0;
    int c=0;
    ifstream f("data_v5.bin",ios::binary);
    ofstream ff("data_v5mod.bin",ios::binary);
    if(!f.is_open()) {
        cerr << "Error: file is not open\n";
        return 1;
    }
    while(f.read((char*)&e,sizeof(double))) {
        c++;
    }
    f.clear();
```

```

        f.seekg( ios::beg );
        cout<<c<<endl;
        double var[c];
        for(int i=0;i<c;i++){
            f.read( (char*)&var[i], sizeof(double) );
        }
        f.close();
        float rev[c];
        for(int i=0;i<c;i++){
            rev[i]=1/var[i];
            ff.write( (char*)&rev[i], sizeof(float) );
        }
        ff.close();
        return 0;
    }

```

4 Вывод

В процессе выполнения лабораторной работы были изучены основные возможности библиотеки ввода-вывода и библиотеки для работы с файлами, было освоено форматированный ввод-вывод, символьный и блочный ввод-вывод и работа с файлами, были получены практические навыки по использованию методов read() и write(), применению манипуляторов setw, hex, oct, dec, классов fstream, ofstream, ifstream.