

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

ОТЧЕТ
о лабораторной работе №5
РАБОТА С ФАЙЛАМИ В СТАНДАРТНОЙ БИБЛИОТЕКИ СИ++

Дисциплина: Языки программирования

Группа: 18ПИ1

Выполнил: Водянов В.О.

Количество баллов:

Дата сдачи:

Принял: 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 Задание к лабораторной работе

2.1 Написать программу, читающую последовательность из десяти целых чисел, а затем выводящих 37 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.

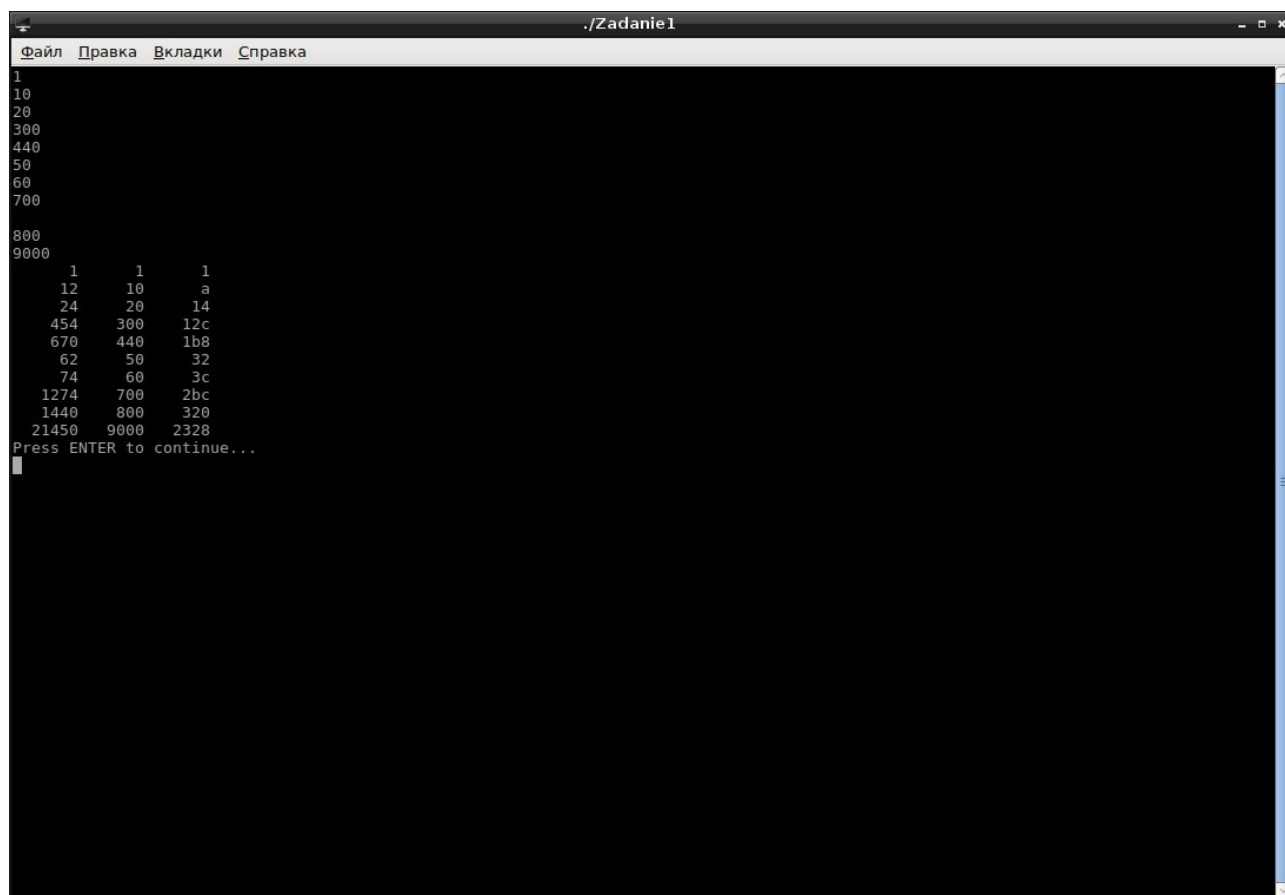
2.4 Написать программу, читающую содержимое одного текстового файла, и записывающую его в конец другого текстового файла. Файлы для работы создать самостоятельно.

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

3 Результаты работы

3.1 Для реализации данной программы использовалось два цикла. В первом цикле вводится последовательность из 10 элементов. Во втором цикле осуществляется вывод массива в 3-х система счисления и с шириной поля в 7 символов. Работа программы представлена на Рисунке 1. Алгоритм работы

программы представлен на Рисунке 2. Полный текст программы представлен в Приложении А.



```
./Zadanie1
Файл  Правка  Вкладки  Справка
1
10
20
300
440
50
60
700
800
9000
      1      1      1
     12     10     a
     24     20     14
    454    300    12c
    670    440    1b8
     62     50     32
     74     60     3c
   1274    700    2bc
   1440    800    320
  21450   9000   2328
Press ENTER to continue...
```

Рисунок 1 - Работа программы вывода последовательности

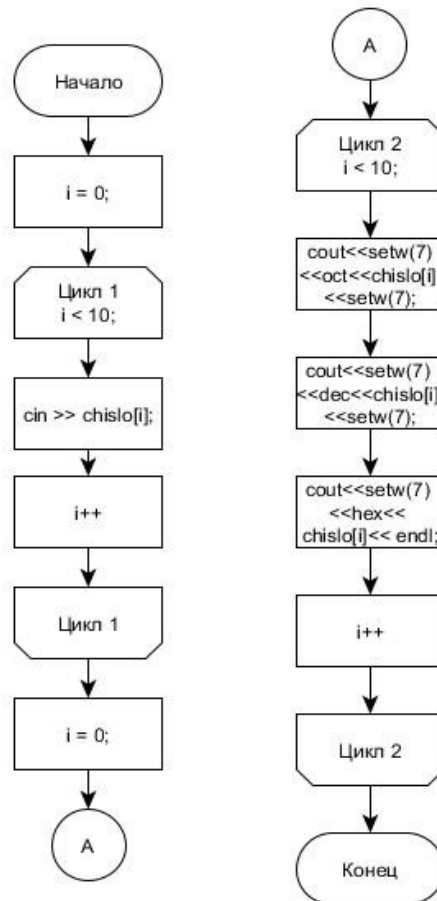


Рисунок 2 - Блок-схема 1

3.2 В данной программе используется функция с вложенными циклами, которая посредством оператора switch выводится таблицу умножения в нужной системе счисления. Работа программы представлена на Рисунке 3. Алгоритм работы функции представлен на Рисунке 4. Полный текст программы представлен в Приложении Б.

```
./Zadanie2
Файл  Правка  Вкладки  Справка
Введите основание таблицы умножения: 16
ТАБЛИЦА УМНОЖЕНИЯ ДЛЯ СИСТЕМЫ С ОСНОВАНИЕМ 16:
 2 * 1 = 2   3 * 1 = 3   4 * 1 = 4   5 * 1 = 5   6 * 1 = 6   7 * 1 = 7   8 * 1 = 8   9 * 1 = 9
 2 * 2 = 4   3 * 2 = 6   4 * 2 = 8   5 * 2 = a   6 * 2 = c   7 * 2 = e   8 * 2 = 10  9 * 2 = 12
 2 * 3 = 6   3 * 3 = 9   4 * 3 = c   5 * 3 = f   6 * 3 = 12  7 * 3 = 15  8 * 3 = 18  9 * 3 = 1b
 2 * 4 = 8   3 * 4 = c   4 * 4 = 10  5 * 4 = 14  6 * 4 = 18  7 * 4 = 1c  8 * 4 = 20  9 * 4 = 24
 2 * 5 = a   3 * 5 = f   4 * 5 = 14  5 * 5 = 19  6 * 5 = 1e  7 * 5 = 23  8 * 5 = 28  9 * 5 = 2d
 2 * 6 = c   3 * 6 = 12  4 * 6 = 18  5 * 6 = 1e  6 * 6 = 24  7 * 6 = 2a  8 * 6 = 30  9 * 6 = 36
 2 * 7 = e   3 * 7 = 15  4 * 7 = 1c  5 * 7 = 23  6 * 7 = 2a  7 * 7 = 31  8 * 7 = 38  9 * 7 = 3f
 2 * 8 = 10  3 * 8 = 18  4 * 8 = 20  5 * 8 = 28  6 * 8 = 30  7 * 8 = 38  8 * 8 = 40  9 * 8 = 48
 2 * 9 = 12  3 * 9 = 1b  4 * 9 = 24  5 * 9 = 2d  6 * 9 = 36  7 * 9 = 3f  8 * 9 = 48  9 * 9 = 51
 2 * a = 14  3 * a = 1e  4 * a = 28  5 * a = 32  6 * a = 3c  7 * a = 46  8 * a = 50  9 * a = 5a
Press ENTER to continue...
█
```

Рисунок 3 - Работа программы таблицы умножения

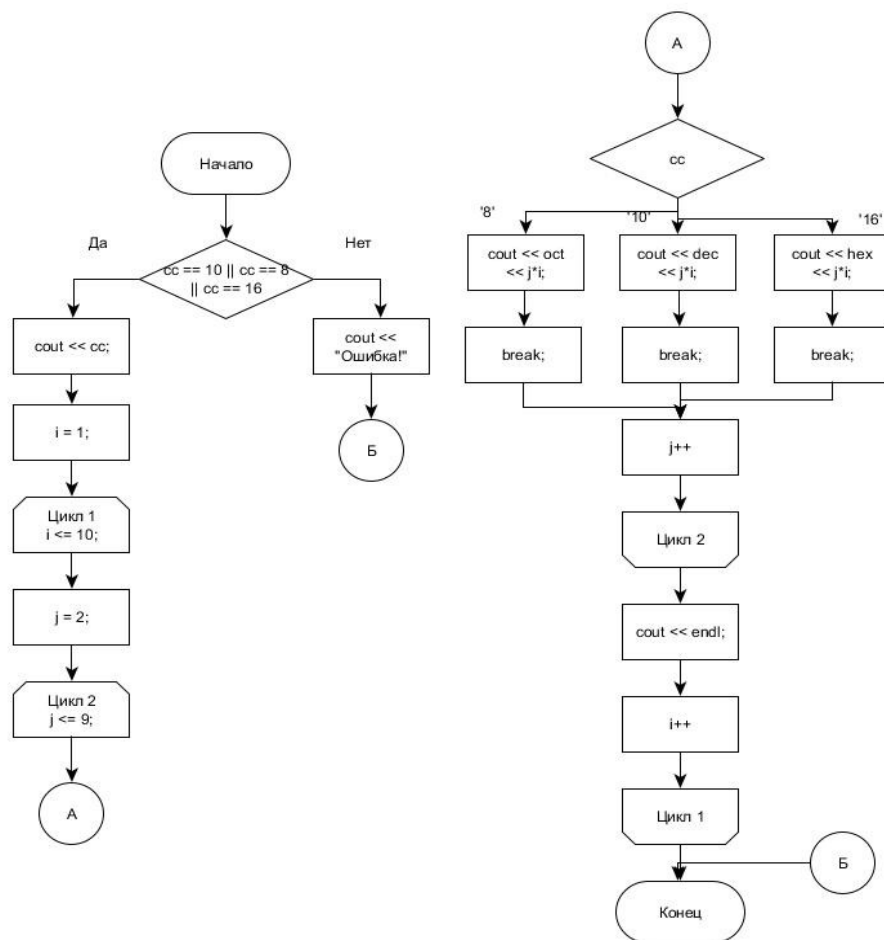


Рисунок 4 - Блок-схема 2

3.3 Нахождение среднего значения осуществляется по формуле sum/count , где sum это сумма всех значение из файла `data_v8.txt`. count это количество этих значений. Работа программы представлена на Рисунке 5. Алгоритм работы программы представлен на Рисунке 6. Полный текст программы представлен в Приложении В.

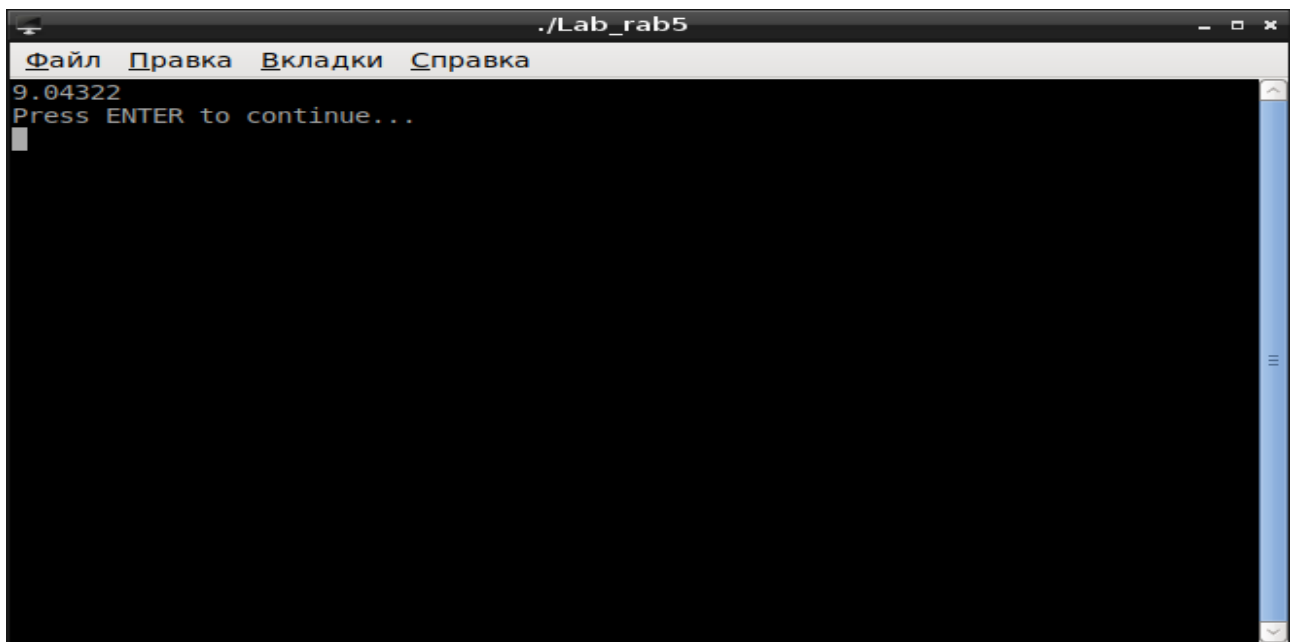


Рисунок 5 - Работа программы для чтения из файла

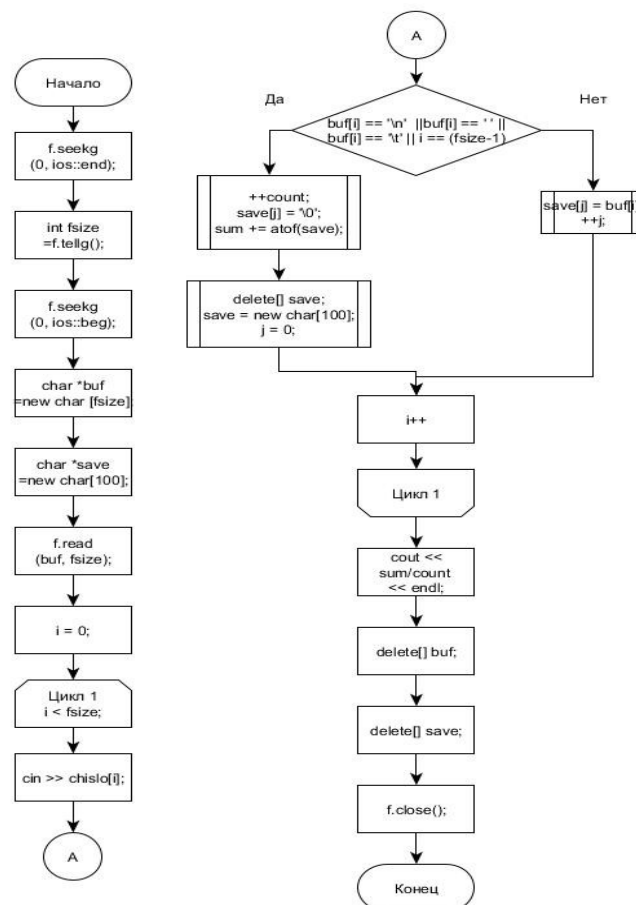


Рисунок 6 - Блок-схема 3

3.4 Программа работает следующим образом. Сначала мы читаем файл *f* в переменную *buf* типа *char**. Затем с помощью метода *write* библиотеки *fstream* мы записываем в другой файл *f2* наш *buf*. Работа программы представлена на Рисунке 7. Алгоритм работы программы представлен на Рисунке 8. Полный текст программы представлен в Приложении Г.

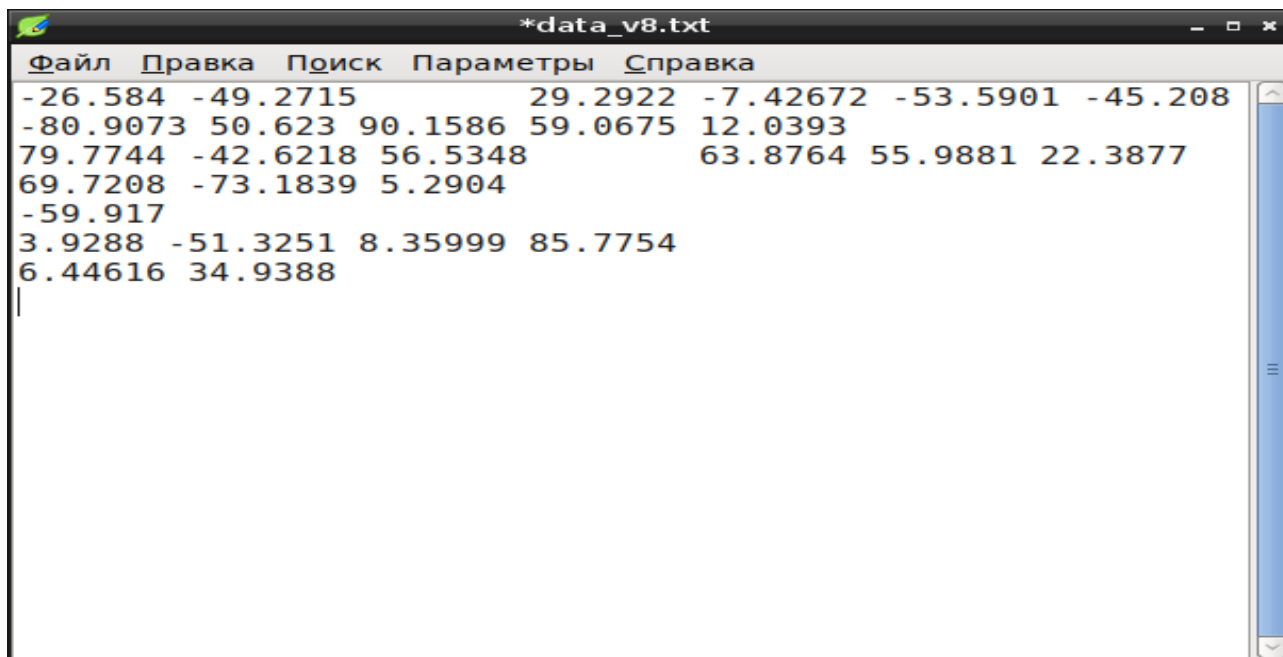


Рисунок 7 - Работа программы для записи в файл

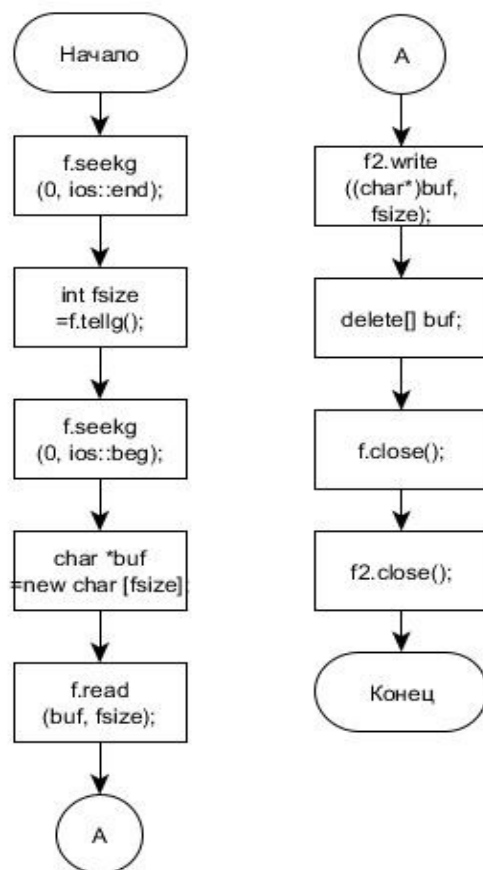


Рисунок 8 - Блок-схема 4

3.5 Программа работает следующим образом. Сначала мы читаем файл `f` в переменную `buf` типа `char*`. В цикле каждому `i` элементу `buf2` типа `char*` присваивается отношение единицы к значению `i` элемента `buf`, преобразованное к типу `float`. Затем с помощью метода `write` библиотеки `fstream` мы записываем в другой файл `f2` наш `buf2`. Работа программы представлена на Рисунке 9. Алгоритм программы представлен на Рисунке 10. Полный текст программы представлен в Приложении Д.

```

файл Правка Вкладки Справка
0.0105263
0.0109492
0.0322581
0.0181818
0.0416667
0.0103934
0.0147059
0.0110988
0.0769231
0.0204082
0.00917431
0.0105263
0.047619
0.0106667
0.0277778
0.025641
0.106667
0.00313089
0.02
0.027037
0.0833333
0.0147059
0.0106383
0.0102941
0.0133846
0.01
0.0238095
0.00990099
0.2
0.0103934
0.0117647
0.0147059
0.0066667
Inf
0.0110279
0.0322581
0.0526316
0.0106667
0.0454545
0.0294118
0.0120295
0.0106667
0.0133846
0.0109492
0.047619
0.0106667
0.00833333
0.008083
0.008
0.0090991
0.0130809
0.0294118
0.142057
0.0106667
Press ENTER to continue...

```

Рисунок 9 - Работа программы для записи в двоичный файл

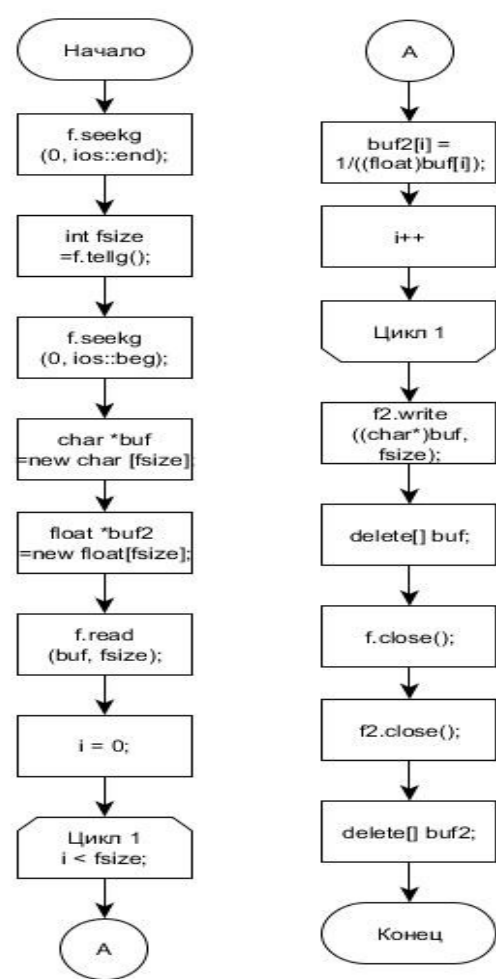


Рисунок 10 - Блок-схема 5

4 Вывод

В результате выполнения работы были изучены операции форматированного и блочного ввода-вывода с помощью библиотеки языка Си++, а также была написана функция для работы с форматированным вводом-выводом, и получены практические навыки в написании программ блочного ввода-вывода на C++.

Приложение А

Текст программы вывода последовательности

```
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char **argv)
{
    int chislo[10], i;
    for (i = 0; i < 10; i++)
    {
        cin >> chislo[i];
    }
    for (i = 0; i < 10; i++)
    {
        cout << setw(7) << oct << chislo[i] << setw(7);
        cout << setw(7) << dec << chislo[i] << setw(7);
        cout << setw(7) << hex << chislo[i] << endl;
    }
    return 0;
}
```

Приложение Б

Текст программы таблицы умножения

```
#include <iostream>
#include <iomanip>
using namespace std;

void table_ymoz(int cc);

int main()
{
    int cc;
    cout << "Введите основание таблицы умножения: ";
    cin >> cc;
    table_ymoz(cc);
    return 0;
}

void table_ymoz(int cc)
{
    int i = 0, j = 0;

    if (cc == 10 || cc == 8 || cc == 16)
    {
        cout << "ТАБЛИЦА УМНОЖЕНИЯ ДЛЯ СИСТЕМЫ С ОСНОВАНИЕМ " << cc <<
": " << endl;
        for(i = 1; i <= 10; i++)
        {
            for(j = 2; j <= 9; j++)
            {
                switch(cc)
                {
```

```

        case 8:  cout << oct << setw(5) << j << setw(1) <<
" * " << setw(2) << left << i << setw(2) << right << "=" <<
setw(4) << j*i; break;

        case 10: cout << dec << setw(5) << j << setw(1) <<
" * " << setw(2) << left << i << setw(2) << right << "=" <<
setw(3) << j*i; break;

        case 16: cout << hex << setw(5) << j << setw(1) <<
" * " << setw(2) << left << i << setw(2) << right << "=" <<
setw(3) << j*i; break;

    }

    }

    cout << endl;
}

}

        else      cout    <<    "Ошибка.    Данное    основание    не
поддерживается!!!"<< endl;
}

```

Приложение В

Текст программы для чтения из файла

```
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;

int main()
{
    int i = 0, j = 0, count = 0;
    double sum = 0.0;
    ifstream f("/root/Laba5/Vodyanov3/data_v8.txt");
    f.seekg(0, ios::end);
    int fsize = f.tellg();
    f.seekg(0, ios::beg);
    char *buf = new char [fsize];
    char *save = new char[100];
    f.read(buf, fsize);
    for (i = 0; i < fsize; i++)
    {
        if (buf[i] == '\n' || buf[i] == ' ' || buf[i] == '\t' ||
i == (fsize-1))
        {
            ++count;
            save[j] = '\0';
            sum += atof(save);
            delete[] save;
            save = new char[100];
            j = 0;
        }
        else
        {
            save[j] = buf[i];
```

```
        ++j;
    }
}
cout << sum/count << endl;
delete[] buf;
delete[] save;
f.close();
return 0;
}
```


Приложение Г

Текст программы для записи в файл

```
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;

int main()
{
    ifstream f("/root/Laba5/Vodyanov3/data_v8.txt");
    ofstream f2("/root/Laba5/Vodyanov3/data_v10.txt", ios::app);
    f.seekg(0, ios::end);
    int fsize = f.tellg();
    f.seekg(0, ios::beg);
    char *buf = new char [fsize];
    f.read(buf, fsize);
    f2.write((char*)buf, fsize);
    delete[] buf;
    f.close();
    f2.close();
    return 0;
}
```

Приложение Д

Текст программы для записи в двоичный файл

```
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;

int main()
{
    ifstream f("/root/Laba5/Vodyanov3/data_v8.bin");
    ofstream f2("/root/Laba5/Vodyanov3/data_v11.bin");
    f.seekg(0, ios::end);
    const int fsize = f.tellg();
    f.seekg(0, ios::beg);
    char *buf = new char[fsize];
    float *buf2 = new float[fsize];
    f.read(buf, fsize);
    int i = 0;
    /*for (i = 0; i < fsize; i++)
        cout << (double)buf[i] << endl;
    */
    for (i = 0; i < fsize; i++)
    {
        buf2[i] = 1/((float)buf[i]);
        cout << buf2[i] << endl;
    }
    f2.write((char*)buf2, fsize);
    f.close();
    f2.close();
    delete[] buf;
    delete[] buf2;
    return 0;
}
```