

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР.

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Программирование»
Тема: «Обработка одномерных массивов».

Студентка гр. 1309

Ищенко Д.О.

Преподаватель

Калмычков В.А.

Санкт-Петербург

2021

Оглавление

Формулировка задания.....	3
Контрольный пример.	3
Математическая постановка задачи.	3
Формат ввода-вывода.....	3
Способ внутреннего хранения данных.....	4
Способ реализации ввода-вывода.....	5
Блок-схема.....	6
Текст программы.	8
Результаты работы программы.	11
Вывод.....	12

Формулировка задания.

Задана последовательность положительных чисел a_1, a_2, \dots, a_n .

Вычислить:

$$\frac{1}{\sqrt{a_1}} + \frac{1}{\sqrt{a_1} + \sqrt{a_2}} + \dots + \frac{1}{\sqrt{a_1} + \sqrt{a_2} + \dots + \sqrt{a_n}}.$$

Контрольный пример.

$$n = 5$$

$$a_1 = 1.4; a_2 = 1.1; a_3 = 45.272; a_4 = 675; a_5 = 0.312796.$$

$$S = 1.46156730\dots$$

Математическая постановка задачи.

Дано: $a_1, a_2, \dots, a_n > 0$

$i = 1, 2, 3, 4, 5 \dots n$

Найти: S

Способ решения:

Считать a_1, a_2, \dots, a_n и вычислить S по формуле. $S =$

$$\frac{1}{\sqrt{a_1}} + \frac{1}{\sqrt{a_1} + \sqrt{a_2}} + \dots + \frac{1}{\sqrt{a_1} + \sqrt{a_2} + \dots + \sqrt{a_n}}.$$

Формат ввода-вывода.

Файлы входных и выходных данных располагается в корневой папке проекта Visual Studio.

Первой строкой входного файла передается целое число n - кол-во чисел, которые должны быть считаны из файла. Далее на каждой новой строке находится число a_i типа double.

В выходном файле в первой строке указывается версия программы (первая версия - с использованием фиксированного массива, вторая – динамического). Далее проверенное число N , совпадающее с реальным количеством считанных элементов. Далее в строке через пробел указываются считанные элементы. Последней строкой выводится искомое S .

$$0 \leq N \leq 9223372036854775807$$

Input.txt	Output.txt
N	program version: 1/2
a_1	actual number of elements: N^1
a_2	$a_1 a_2 \dots a_n$
...	$S = S$
a_n	

Способ внутреннего хранения данных.

Имя переменной	Тип переменной	Ограничения на размер данных	Назначение
f_in	fstream		Хранит входные данные
f_out	fstream		Хранит выходные данные
log	fstream		Содержит информацию о ходе работы программы.
a	double	1,7E +/- 308 (15 знаков)	Хранит один элемент a_i , предназначена для считывания элементов.
S	double		Хранит искомое значение суммы.
s	double		Хранит сумму корней всех i считанных элементов.
n	long long	-9 223 372 036 854 775 808 до +9 223 372 036	Хранит количество элементов в

		854 775 807, целый тип	последовательности (до и после проверки)
--	--	------------------------	--

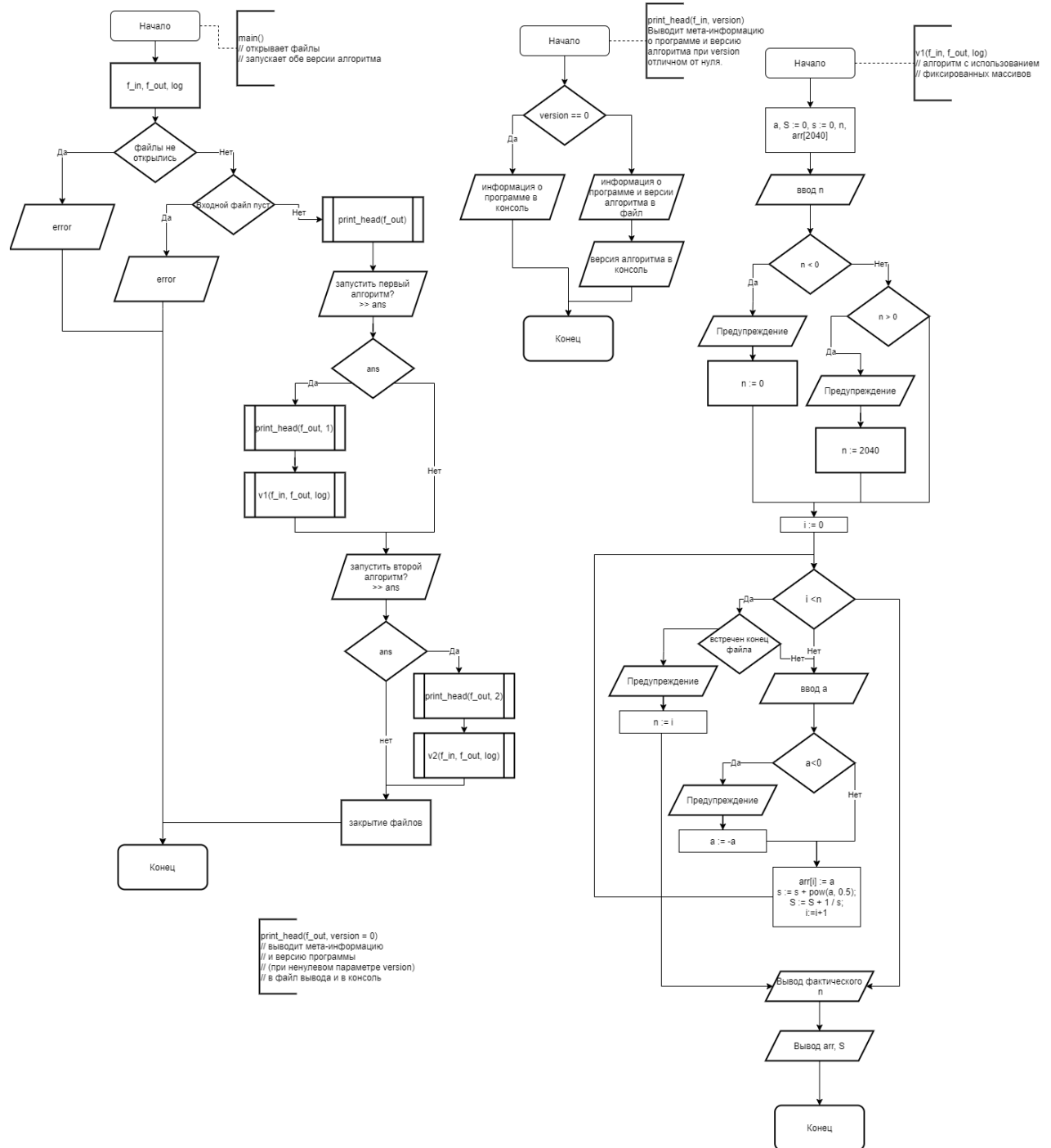
Способ реализации ввода-вывода.

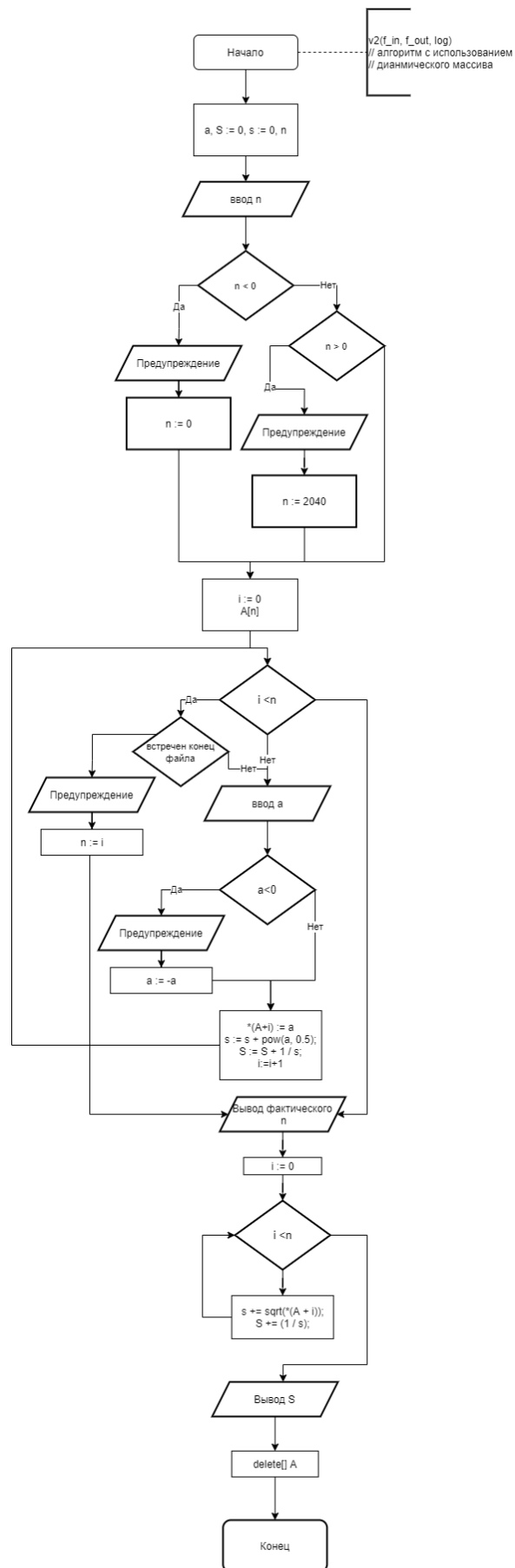
Для ввода-вывода при реализации задания был задействован класс `fstream` из одноименной библиотеки для работы с файлами.

Класс `fstream` позволяет создать объект, с помощью которого возможно записывать и считывать данные из файла. Конструктору класса первым параметром передается расположение и название файла, вторым маркер, означающий тип работы с файлом (чтение/запись).

Для файлов записи (`f_out` и `log`) были задействованы флаги `trunc` и `out`. `Trunc` очищает содержимое файла при его открытии, `out` указывает на то, что файл открыт только для записи.

Блок-схема.





Текст программы.

```
/*
Автор: Ищенко Д.О
Группа 1309
Версия: 4.1.1
Дата начала: 1.11.2021
Дата конца:
Задание: You are given a sequence of positive numbers a1, a2,..., an.
Calculate:
```

$$S = \frac{1}{\sqrt{a_1}} + \frac{1}{\sqrt{a_1} + \sqrt{a_2}} + \dots + \frac{1}{\sqrt{a_1} + \sqrt{a_2} + \dots + \sqrt{a_n}}$$

```
*/
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int v1(fstream& f_in, fstream& f_out, fstream& log);
int v2(fstream& f_in, fstream& f_out, fstream& log);
void print_head(fstream& f_out, int version=0);

int main() {
    string input_path = "../test_in.txt",
           output_path = "../output.txt",
           log_path = "../log.txt";

    fstream f_in(input_path),
             f_out(output_path, fstream::trunc | fstream::out),
             log(log_path, fstream::trunc | fstream::out);

    if (!log.is_open()) {
        cout << "log file open error";
        return -1;
    }
    if (!f_in.is_open() || !f_out.is_open()) {
        log << "files open error" << endl;
        return -1;
    }

    if (f_in.eof()) {
        cout << "no item count was passed";
        log << "no item count was passed" << endl;
        f_out << "no item count was passed";
        return -1;
    }

    print_head(f_out);

    bool ans = false;
    cout << "Start first version? (1/0) : ";
    cin >> ans;
    if (ans) {
        print_head(f_out, 1);
        v1(f_in, f_out, log);

        log.close(); f_in.close(); f_out.close();
        f_in.open(input_path);
    }
}
```



```

        f_out.open(output_path, fstream::trunc | fstream::out);
        log.open(log_path, fstream::trunc | fstream::out);
    }

    cout << "\nStart second version? (1/0) : ";
    cin >> ans;
    if (ans) {
        print_head(f_out, 2);

        v2(f_in, f_out, log);
    }

    log.close(); f_in.close(); f_out.close();

    return 0;
}

void print_head(fstream &f_out, int version) {
    if (version == 0) {
        cout << "Author: Ishchenko D.O.\n"
            << "Group 1309\n"
            << "Version: 4.1.1\n"
            << "Start date : 1.11.2021\n"
            << "End date : \n"
            << "Exercise: You are given a sequence of positive numbers a1,
a2,..., an.\n"

            << "Calculate:\n"
            << "      1      1      1\n"
            << "S = ___ + ___ + ... + ___\n"
            << "      Va1   Va1 + Va2   Va1 + Va2 + ... Van\n"
            << "      Va1   Va1 + Va2   Va1 + Va2 + ... Van\n" << endl;
    }
    else {
        f_out << "Author: Ishchenko D.O.\n"
            << "Group 1309\n"
            << "Version: 4.1.1\n"
            << "Start date : 1.11.2021\n"
            << "End date : \n"
            << "Exercise: You are given a sequence of positive numbers a1, a2,
. . . , an.\n"

            << "Calculate:\n"
            << "      1      1      1\n"
            << "S = ___ + ___ + ... + ___\n"
            << "      Va1   Va1 + Va2   Va1 + Va2 + ... Van\n"
            << "      Va1   Va1 + Va2   Va1 + Va2 + ... Van\n";
        cout << "program version: " << version << endl;
        f_out << "program version: " << version << endl;
    }
}

void v1(fstream& f_in, fstream& f_out, fstream& log)
{
    double a, S = 0, s = 0;
    long long n;
    double arr[2040];

    // get and chek n
    f_in >> n;
    if (n < 0) {
        n = 0;
        log << "number of items has been converted from " << n << " to 0" <<
endl;

```

```

    }
    else if (n > 2040) {
        cout << "the number of elements is too large. 2040 items will be read";
        log << "the number of elements is too large. 2040 items will be read";
        n = 2040;
    }

    //read arr, correct n
    for (int i = 0; i < n; i++) {
        if (f_in.eof()) {
            n = i;
            log << "End of file detected, new value n =" << n << endl;
            break;
        }
        f_in >> a;
        if (a < 0) {
            log << "negative value a = " << a << " on " << i + 1 << " line
converted to positive" << endl;
            a = -a;
        }
        arr[i] = a;
        s += pow(a, 0.5);
        S += 1 / s;
    }
    f_out << "actual number of elements: " << n << endl;
    log << n << " numbers read " << endl;

    for (int i = 0; i < n; i++) {
        f_out << arr[i] << " ";
    }
    log << "items written to file " << endl;
    f_out << endl << "S = " << S;
    cout << "S = " << S;
    log << "S value written to file " << endl;
}

int v2(fstream& f_in, fstream& f_out, fstream& log)
{
    double a, S = 0, s = 0;
    long long n;

    // Get n
    f_in >> n;
    if (n < 0) {
        n = 0;
        log << "number of items has been converted from " << n << " to 0" <<
endl;
    }
    for (int i = 0; i < n; i++) {
        if (f_in.eof()) {
            n = i;
            log << "End of file detected, new value n =" << n << endl;
            break;
        }
        f_in >> a;
    }
    f_in.seekg(std::ios::beg);
    f_in >> a; // move pointer first a (a = n now)
    f_out << "actual number of elements: " << n << endl;

```

```

// define A items
double* A = new double[n];
for (int i = 0; i < n; i++) {
    f_in >> a;
    if (a < 0) {
        log << "negative value a = " << a << " on " << i + 1 << " line
converted to positive" << endl;
        a = -a;
    }
    *(A+i) = a;
    f_out << a << " ";
}
log << n << " numbers read " << endl;
log << "Numbers written to file " << endl;

// calculate S
for (int i = 0; i < n; i++) {
    s += sqrt(*(A + i));
    S += (1 / s);
}

cout << "S = " << S;
f_out << endl << "S = " << S;
log << "S value written to file " << endl;

delete[] A;

return 0;
}

```

Результаты работы программы.

Author: Ishchenko D.O.

Group 1309

Version: 4.1.1

Start date : 1.11.2021

End date :

Exercise: You are given a sequence of positive numbers a_1, a_2, \dots, a_n .

Calculate:

$$S = \frac{1}{a_1} + \frac{1}{a_1 + a_2} + \dots + \frac{1}{a_1 + a_2 + \dots + a_n}$$

$$\frac{1}{a_1} + \frac{1}{a_1 + a_2} + \dots + \frac{1}{a_1 + a_2 + \dots + a_n}$$

program version: 1

actual number of elements: 5

1.4 1.1 45.272 675 0.312796

$S = 1.46157$

Author: Ishchenko D.O.

Group 1309

Version: 4.1.1

Start date : 1.11.2021

End date :

Exercise: You are given a sequence of positive numbers a_1, a_2, \dots, a_n .

Calculate:

$$S = \frac{1}{Va_1} + \frac{1}{Va_1 + Va_2} + \dots + \frac{1}{Va_1 + Va_2 + \dots + Va_n}$$

program version: 2

actual number of elements: 5

1.4 1.1 45.272 675 0.312796

$S = 1.46157$

Вывод.

В ходе выполнения лабораторной работы была написана программа, решающая поставленную задачу. Во время её разработки были получены практические навыки работы с различными конструкциями языка `c++`, такими как инструменты взаимодействия с файлами, динамическими и статическими массивами.