**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №4**

**по дисциплине «Программирование»**

Тема: Программа на С++ сортирующая 2 файла в 1 массив

|  |  |  |
| --- | --- | --- |
| Студент гр. 4352 |  | Редкокаша Р. Ю. |
| Преподаватель |  | Калмычков В. А. |

Санкт-Петербург

2024

**Оглавление**

[1 Исходная формулировка задания 3](#_Toc181896491)

[2 Определение неясностей 3](#_Toc181896492)

[3 Математическая постановка 3](#_Toc181896493)

[4 Устройство файла 3](#_Toc181896494)

[5 Контрольный пример 3](#_Toc181896495)

[6 Ограничения 4](#_Toc181896496)

[7 Организация интерфейса пользователя 4](#_Toc181896497)

[8 Реализация ввода-вывода 4](#_Toc181896498)

[9 Внутреннее представление данных 4](#_Toc181896499)

[10 Описание алгоритма 5](#_Toc181896500)

[11 Тексты программ 6](#_Toc181896501)

[12 Результаты работы программ 9](#_Toc181896502)

[Выводы 12](#_Toc181896503)

**Цель работы.**

Написание программы, получающую на вход значения из файлов, обрабатывающая их требуемым образом и выводящая их в другие файлы.

# Исходная формулировка задания

Даны две последовательности значений *a1 ≤ a2 ≤... ≤ an* и *b1 ≤ b2 ≤ ... ≤ bm*. Сформировать из них последовательность значений *с1 ≤ с2 ≤ ... ≤ сn+m*. Первым числом в файле идёт число значений.

# Определение неясностей

Программа должна считать все значения из файла, вне зависимости от числа, объявленного в файле. Отрицательное число значений принято брать равным нулю. Если заявлено больше значений – программа считает только существующие. По условию задачи – все данные в файлах должны быть указаны в порядке возрастания.

# Математическая постановка

Дано: два файла с числами в порядке возрастание.

Нужно: отсортировать в один массив.

Числа будут сортироваться по их реальному значению, а не по модулю.

# Устройство файла

Файл в первой строке содержит число – количество обрабатываемых значений, в последующих вещественные числа в неубывания. Выходной файл содержит числа в порядке неубывания. При определении количества значений как отрицательное – количество будет считать равным нулю.

# Контрольный пример

*Ввод:*

|  |  |  |
| --- | --- | --- |
| Файл | A | B |
| 1 | -872.3420236 | -766.4065798 |
| 2 | -816.2110490 | -753.7560521 |
| 3 | -750.1402855 | -745.0527623 |
| 4 | -738.0068281 | -702.5387958 |
| 5 | -733.6132837 | -489.4239603 |
| … | … | … |
| 21 | 442.4667908 | 508.9143389 |
| 22 | 469.8615767 | 717.3395407 |
| 23 | 699.8609559 | 810.5816263 |
| 24 | 844.7001862 | 830.8404718 |
| 25 | 919.8795779 | 900.2240844 |

*Вывод:*

|  |  |
| --- | --- |
| 1 | -872.3420236 |
| 2 | -816.2110490 |
| 3 | -766.4065798 |
| 4 | -753.7560521 |
| 5 | -750.1402855 |
| … | … |
| 25 | -150.2098076 |
| … | … |
| 46 | 810.5816263 |
| 47 | 830.8404718 |
| 48 | 844.7001862 |
| 49 | 900.2240844 |
| 50 | 919.8795779 |

# Ограничения

Вводимые данные ограничены ёмкостью double.

# Организация интерфейса пользователя

Данные отображаются в программной консоли и файле в формате текста и знаков. Для чисел отображается 7 знаков после запятой.

|  |  |  |
| --- | --- | --- |
| № | Файл | Содержимое |
| О1 | Fo | Задание:  Даны две последовательности значений a1 ≤ a2 ≤ ... ≤ an и b1 ≤ b2 ≤ ... ≤ bm.  Сформировать из них последовательность значений с1 ≤ с2 ≤ ... ≤ сn+m.  Автор: Редкокаша Роман Юрьевич Версия: 4.1  Группа: 4352  Даты: начало: 23.10.2024 - конец: 23.10.2024 |
| О2 | Файл name.txt: |
| O3 | Выходной файл: |
| O4 |  |
| E1 | Ferror  Fo | Файл Name.txt не помещён в корневой каталог программы. |
| E2 | Файл Name.txt не содержит значений. |
| E3 | Длина файла больше 100, что дальше этого значения - учитываться не будет! |
| E4 | Значение чисел отрицательно, было приравнено к нулю. |
| E5 | Требуемая длинна больше файла! Будет приравнено к настоящей длине. |

# Реализация ввода-вывода

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Ввод | Вывод | Дробная часть |
| iostream | cin | cout |  |
| iomanip |  |  | cout.setf(ios::fixed);  cout.precision(7); |
| fstream | ios::in >> | ios::out << |  |

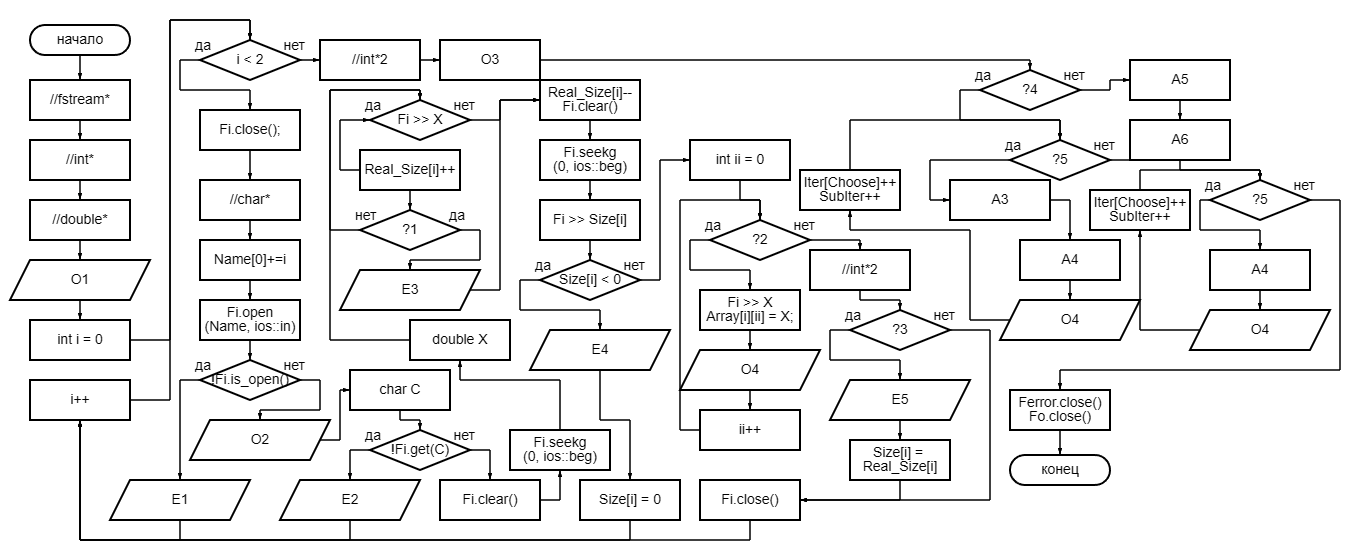
# Внутреннее представление данных

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Назначение |
| С | char | Проверка на существование данных в файле |
| Fi | fstream[2] | Определение файлов на ввод |
| Fo | fstream | Создание файла вывода |
| Ferror | Создания файла вывода ошибок |
| SumSize | int | Длина выходного массива |
| Size | Хранение указанной длины |
| Choose | Вспомогательная переменная для сортировки |
| Iter | int[2] |
| Real\_Size | Хранение реального объёма |
| X | double | Файл для временного хранения данных из файла |
| Array | double\*\*double\*[2]double[Real\_Size[2]] | Массив, создающийся из файла |
| double[2][100] |
| ArrayOut | double\*[SumSize] | Итоговый массив |
|  | Double[200] |

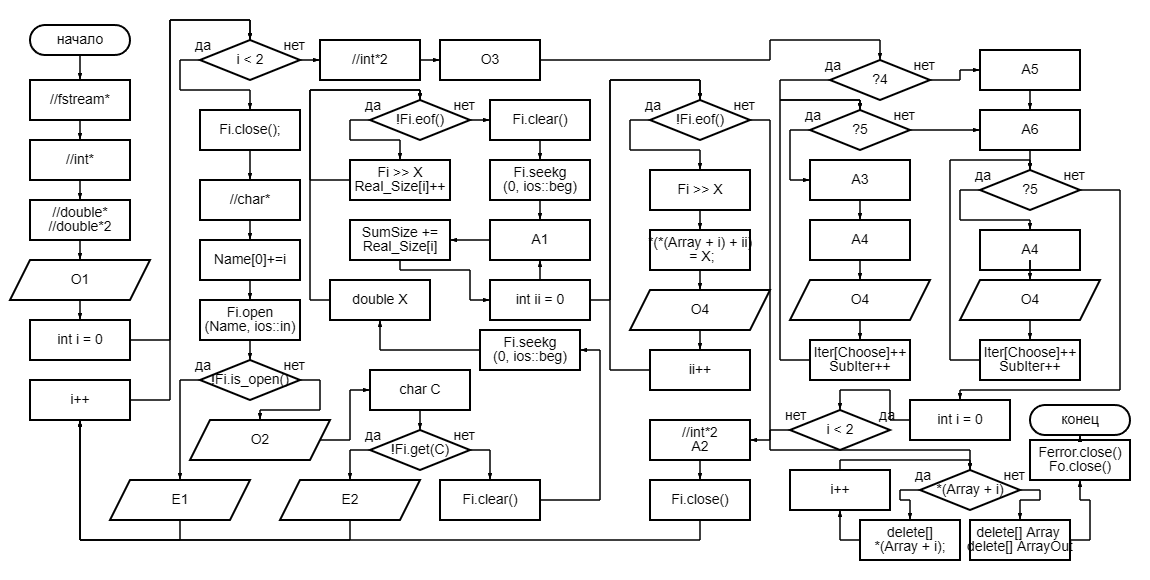
# Описание алгоритма

Программа проверяет наличие 2-х файлов с именами «A.txt» и «B.txt», с числами в порядке возрастания, в корневой папке программы и пуст ли файл, если да, то уведомляет об этом. Если в одном из файлов длина указанная больше реальной, то она приравнивается к реальной. Дальше считывает данные из файлов в массивы, создаёт массив с этими числами в порядке возрастания.

*Версия 1.*



*Версия 2.*

**

*\*//fstream; fstream Fi, Fo("out.txt", ios::out), Ferror("error.txt", ios::out)*

*//double: 1) double Array[2][100], ArrayOut[200]*

*2)* *double\*\* Array = new double\* [2] {nullptr, nullptr}*

*//double2: 2) double ArrayOut = nullptr*

*//char: char Name[] = "A.txt"*

*//int: int Choose = 0, Iter[2]{ 0,0 }, SubIter = 0*

*//int2: int Choose = 0, Iter[2]{ 0,0 }, SubIter = 0*

*?1: 1) Real\_Size[i] > 100*

*?2: 1) !Fi.eof() && ii < Real\_Size[i]*

*?3* 1) *Size[i] > Real\_Size[i]*

*?4: Real\_Size[0] != 0 && Real\_Size[1] != 0*

*?5: 1) Iter[Choose] < Size[Choose]*

*2)* *Iter[Choose] < Real\_Size[Choose]*

*A1: 2) \*(Array + i) = new double[Real\_Size[i]]*

*A2: 2) ArrayOut = new double[SumSize]*

*A3: 1) Choose = (Array[0][Iter[0]] < Array[1][Iter[1]]) ? 0 : 1*

*2) Choose = (\*(\*(Array + 0) + Iter[0]) < \*(\*(Array + 1) + Iter[1])) ? 0 : 1*

*A4: 1)* *ArrayOut[SubIter] = Array[Choose][Iter[Choose]]*

*2) \*(ArrayOut + SubIter) = \*(\*(Array + Choose) + Iter[Choose])*

*A5: Choose = (Real\_Size[0] < Real\_Size[1]) ? 0 : 1*

*A6: Choose = (Choose == 1) ? 0 : 1*

# Тексты программ

Версия №1

/\*я—»√ћјя –”“ќ…я—»√ћј

\* Задание:

\* Даны две последовательности значений a1 ≤ a2 ≤ ... ≤ an и b1 ≤ b2 ≤ ... ≤ bm.

\* Сформировать из них последовательность значений с1 ≤ с2 ≤ ... ≤ сn+m.

\* Автор: Редкокаша Роман Юрьевич Версия: 4.1

\* Группа: 4352

\* Даты: начало: 23.10.2024 - конец: 23.10.2024

\*/

#include "iostream"

#include "iomanip"

#include "fstream"

using namespace std;

int main() {

fstream Fi, Fo("out.txt", ios::out), Ferror("error.txt", ios::out); //fstream\*

int Size[2]{ 0, 0 }, Real\_Size[2]{ 0, 0 }; //int\*

double Array[2][100], ArrayOut[200]; //double\*

setlocale(LC\_ALL, "RUSSIAN");

cout.setf(ios::fixed);

cout.precision(7);

cout <<"Задание:\n"<<"\tДаны две последовательности значений a1 <= a2 <= ... <= an и b1 <= b2 <= ... <= bm.\n"<<"\tСформировать из них последовательность значений с1 <= с2 <= ... <= сn + m.\n\n"<<"Автор: Редкокаша Роман Юрьевич Версия: 4.1\n"<<"Группа: 4352\n" <<"Даты: начало: 23.10.2024 - конец: 23.10.2024\n" << endl;

Fo <<"Задание:\n"<<"\tДаны две последовательности значений a1 <= a2 <= ... <= an и b1 <= b2 <= ... <= bm.\n"<<"\tСформировать из них последовательность значений с1 <= с2 <= ... <= сn + m.\n\n"<<"Автор: Редкокаша Роман Юрьевич Версия: 4.1\n"<<"Группа: 4352\n"<<"Даты: начало: 23.10.2024 - конец: 23.10.2024\n" << endl;

for (int i = 0; i < 2; i++) {

Fi.close();

char Name[] = "A.txt"; //char\*

Name[0] += i;

Fi.open(Name, ios::in);

if (!Fi.is\_open()) {

cout << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

Ferror << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

Fo << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

continue;

}

cout << "\n\tФайл " << Name << ".txt: " << endl;

Fo << "\n\tФайл " << Name << ".txt: " << endl;

char C;

if (!Fi.get(C)) {

cout << "Файл " << Name << " не содержит значений." << endl;

Ferror << "Файл " << Name << " не содержит значений." << endl;

Fo << "Файл " << Name << " не содержит значений." << endl;

continue;

}

Fi.clear();

Fi.seekg(0, ios::beg);

double X;

while (Fi >> X) {

Real\_Size[i]++;

if (Real\_Size[i] > 100) { //?1

cout << "Длина файла больше 100, что дальше этого значения - учитываться не будет!" << endl;

Ferror << "Длина файла больше 100, что дальше этого значения - учитываться не будет!" << endl;

Fo<<"Длина файла больше 100, что дальше этого значения - учитываться не будет!"<<endl;

break;

}

}

Real\_Size[i]--;

Fi.clear();

Fi.seekg(0, ios::beg);

Fi >> Size[i];

if (Size[i] < 0)

{

cout << "Значение чисел отрицательно, было приравнено к нулю." << endl;

Ferror << "Значение чисел отрицательно, было приравнено к нулю." << endl;

Fo << "Значение чисел отрицательно, было приравнено к нулю." << endl;

Size[i] = 0;

continue;

}

for (int ii = 0; !Fi.eof() && ii < Real\_Size[i]; ii++) { //?2

Fi >> X;

Array[i][ii] = X;

cout << Array[i][ii] << endl;

Fo << Array[i][ii] << endl;

}

if (Size[i] > Real\_Size[i]) { //?3

cout << "Требуемая длинна больше файла! Будет приравнено к настоящей длине." << endl;

Ferror << "Требуемая длинна больше файла! Будет приравнено к настоящей длине."<< endl;

Fo << "Требуемая длинна больше файла! Будет приравнено к настоящей длине." << endl;

Size[i] = Real\_Size[i];

}

}

Fi.close();

int Choose = 0, Iter[2]{ 0,0 }, SubIter = 0; //int\*2

cout << "\n\tВыходной файл:" << endl;

Fo << "\n\tВыходной файл:" << endl;

if (Real\_Size[0] != 0 && Real\_Size[1] != 0)

for (; Iter[Choose] < Size[Choose]; Iter[Choose]++, SubIter++) { //?4

Choose = (Array[0][Iter[0]] < Array[1][Iter[1]]) ? 0 : 1; //A1

ArrayOut[SubIter] = Array[Choose][Iter[Choose]]; //A2

Fo << ArrayOut[SubIter] << endl;

cout << ArrayOut[SubIter] << endl;

}

else

Choose = (Real\_Size[0] < Real\_Size[1]) ? 0 : 1;

Choose = (Choose == 1) ? 0 : 1; //A3

for (; Iter[Choose] < Size[Choose]; Iter[Choose]++, SubIter++) { //?4

ArrayOut[SubIter] = Array[Choose][Iter[Choose]]; // A2

Fo << ArrayOut[SubIter] << endl;

cout << ArrayOut[SubIter] << endl;

}

Ferror.close();

Fo.close();

return 0;

}

Версия №2

/\*я—»√ћјя –”“ќ…я—»√ћј

\* Задание:

\* Даны две последовательности значений a1 ≤ a2 ≤ ... ≤ an и b1 ≤ b2 ≤ ... ≤ bm.

\* Сформировать из них последовательность значений с1 ≤ с2 ≤ ... ≤ сn+m.

\* Автор: Редкокаша Роман Юрьевич Версия: 4.1

\* Группа: 4352

\* Даты: начало: 23.10.2024 - конец: 23.10.2024

\*/

#include "iostream"

#include "iomanip"

#include "fstream"

using namespace std;

int main() {

fstream Fi, Fo("out.txt", ios::out), Ferror("error.txt", ios::out); //fstream\*

int Real\_Size[2]{ 0, 0 }, SumSize = 0; //int\*

double\*\* Array = new double\* [2] {nullptr, nullptr}; //double\*

double\* ArrayOut = nullptr ; //double\*\*

setlocale(LC\_ALL, "RUSSIAN");

cout.setf(ios::fixed);

cout.precision(7);

cout <<"Задание:\n"<<"\tДаны две последовательности значений a1 <= a2 <= ... <= an и b1 <= b2 <= ... <= bm.\n"<<"\tСформировать из них последовательность значений с1 <= с2 <= ... <= сn + m.\n\n"<<"Автор: Редкокаша Роман Юрьевич Версия: 4.1\n"<<"Группа: 4352\n" <<"Даты: начало: 23.10.2024 - конец: 23.10.2024\n" << endl;

Fo <<"Задание:\n"<<"\tДаны две последовательности значений a1 <= a2 <= ... <= an и b1 <= b2 <= ... <= bm.\n"<<"\tСформировать из них последовательность значений с1 <= с2 <= ... <= сn + m.\n\n"<<"Автор: Редкокаша Роман Юрьевич Версия: 4.1\n"<<"Группа: 4352\n"<<"Даты: начало: 23.10.2024 - конец: 23.10.2024\n" << endl;

for (int i = 0; i < 2; i++) {

Fi.close();

char Name[] = "A.txt";

Name[0] += i;

Fi.open(Name, ios::in);

if (!Fi.is\_open()) {

cout << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

Ferror << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

Fo << "Файл " << Name << " не помещён в корневой каталог программы." << endl;

continue;

}

cout << "\n\tФайл " << char('A' + i) << ".txt: " << endl;

Fo << "\n\tФайл " << char('A' + i) << ".txt: " << endl;

char C;

if (!Fi.get(C)) {

cout << "Файл " << Name << " не содержит значений." << endl;

Ferror << "Файл " << Name << " не содержит значений." << endl;

Fo << "Файл " << Name << " не содержит значений." << endl;

continue;

}

Fi.clear();

Fi.seekg(0, ios::beg);

double X;

while (!Fi.eof()) {

Fi >> X;

Real\_Size[i]++;

}

Fi.clear();

Fi.seekg(0, ios::beg);

\*(Array + i) = new double[Real\_Size[i]]; // A1

SumSize += Real\_Size[i];

for (int ii = 0; !Fi.eof(); ii++) { //?2

Fi >> X;

\*(\*(Array + i) + ii) = X;

cout << \*(\*(Array + i) + ii) << endl;

Fo << \*(\*(Array + i) + ii) << endl;

}

}

Fi.close();

int Choose = 0, Iter[2]{ 0,0 }, SubIter = 0; //int\*2

ArrayOut = new double[SumSize]; //A2

cout << "\n\tВыходной файл:" << endl;

Fo << "\n\tВыходной файл:" << endl;

if (Real\_Size[0] != 0 && Real\_Size[1] != 0)

for (; Iter[Choose] < Real\_Size[Choose]; Iter[Choose]++, SubIter++) { //?4

Choose = (\*(\*(Array + 0) + Iter[0]) < \*(\*(Array + 1) + Iter[1])) ? 0 : 1; //A3

\*(ArrayOut + SubIter) = \*(\*(Array + Choose) + Iter[Choose]); //A4

Fo << \*(ArrayOut + SubIter) << endl;

cout << \*(ArrayOut + SubIter) << endl;

}

else

Choose = (Real\_Size[0] < Real\_Size[1]) ? 0 : 1;

Choose = (Choose == 1) ? 0 : 1; //A5

for (; Iter[Choose] < Real\_Size[Choose]; Iter[Choose]++, SubIter++) {

\*(ArrayOut + SubIter) = \*(\*(Array + Choose) + Iter[Choose]); //A4

Fo << \*(ArrayOut + SubIter) << endl;

cout << \*(ArrayOut + SubIter) << endl;

}

for (int i = 0; i < 2; i++)

if (\*(Array + i))

delete[] \*(Array + i);

delete[] Array;

delete[] ArrayOut;

Ferror.close();

Fo.close();

return 0;

}

# Результаты работы программ

Версия №1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Файл A | Файл B | Файл out | | Файл error |
| 25  -872.342024  -816.211049  -750.140286  -738.006828  -733.613284  -707.381750  -578.355680  -549.047796  -423.828678  -380.807573  -326.860956  -326.472998  -256.599007  -99.068901  -97.158908  -23.085661  -3.865301  32.795158  223.354438  282.936685  442.466791  469.861577  699.860956  844.700186  919.879578 | 25  -766.4065798  -753.7560521  -745.0527623  -702.5387958  -489.4239603  -479.8851645  -369.4357542  -327.4773433  -256.9404097  -240.0086903  -160.3258845  -150.2098076  -133.3743017  -10.5456238  31.1818746  77.8206083  104.8361266  126.6716325  171.5909373  433.4512725  508.9143389  717.3395407  810.5816263  830.8404718  900.2240844 | Задание:  Даны две последовательности значений a1 <= a2 <= ... <= an и b1 <= b2 <= ... <= bm.  Сформировать из них последовательность значений с1 <= с2 <= ... <= сn + m.  Автор: Редкокаша Роман Юрьевич Версия: 4.1  Группа: 4352  Даты: начало: 23.10.2024 - конец: 23.10.2024  Файл A.txt.txt:  -872.342  …  919.88  Файл B.txt.txt:  -766.407  …  900.224  Выходной файл:  -872.342  -816.211  -766.407  -753.756  -750.14  …  810.582  830.84  844.7  900.224  919.88 | |  |
| 0  -872.342024  …  919.879578 | 25  -766.4065798  …  900.2240844 | …  Файл A.txt.txt:  -872.342  …  919.88  Файл B.txt.txt:  -766.407  ...  900.224  Выходной файл:  -766.407  -753.756  -745.053  -702.539  -489.424  …  508.914  717.34  810.582  830.84  900.224 | |  |
| 25  -872.342024  …  919.879578 | 250  -766.4065798  …  900.2240844 | Файл A.txt.txt:  -872.342  …  919.88  Файл B.txt.txt:  -766.407  …  900.224  Требуемая длинна больше файла! Будет приравнено к настоящей длине.  Выходной файл:  -872.342  -816.211  -766.407  -753.756  -750.14  …  830.84  844.7  900.224  900.224  919.88 | | Требуемая длинна больше файла! Будет приравнено к настоящей длине. |
| нету | -25  -766.4065798  …  900.2240844 | Файл A.txt не помещён в корневой каталог программы.  Файл B.txt.txt:  Значение чисел отрицательно, было приравнено к нулю.  Выходной файл: | Файл A.txt не помещён в корневой каталог программы.  Значение чисел отрицательно, было приравнено к нулю. | |
| 25 |  | Файл A.txt.txt:  Требуемая длинна больше файла! Будет приравнено к настоящей длине.  Файл B.txt.txt:  Файл B.txt не содержит значений.  Выходной файл: | | Требуемая длинна больше файла! Будет приравнено к настоящей длине.  Файл B.txt не содержит значений. |

Версия №2

|  |  |  |  |
| --- | --- | --- | --- |
| -872.342024  …  919.879578 | -766.4065798  …  900.2240844 | Файл A.txt::  -872.342  …  919.88  Файл B.txt:  -766.407  …  900.224  Выходной файл:  -872.342  -816.211  -766.407  -753.756  -750.14  …  844.7  900.224  900.224  919.88  919.88 |  |
|  | -766.4065798  …  900.2240844 | Файл A.txt::  Файл A.txt не содержит значений.  Файл B.txt:  -766.407  …  900.224  Выходной файл:  -766.407  -753.756  -745.053  -702.539  -489.424  …  717.34  810.582  830.84  900.224  900.224 | Файл A.txt не содержит значений. |
| -872.342024  …  919.879578 | нету | Файл A.txt::  -872.342  …  919.88  Файл B.txt не помещён в корневой каталог программы.  Выходной файл:  -872.342  -816.211  -750.14  -738.007  -733.613  …  469.862  699.861  844.7  919.88  919.88 | Файл B.txt не помещён в корневой каталог программы. |

Выводы.

Написал программу, считывающие числа из файлов, производящую требуемые действия с данными и записывающая все числа в другой файл. Научился работать с массивами и динамической памятью. Обнаружил, что вещественные числа в документе, также как и в консоли, должны записываться через разделитель – точка.