**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №4**

**по дисциплине «Программирование»**

Тема: **Использование файлов для ввода-вывода данных. Реализация программы для обработки последовательности значений с использованием динамического массива**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2302 |  | Коняев А.Е. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2022

**Оглавление**

[**Исходная формулировка задачи** 3](#_Toc118197897)

[**Анализ задачи и устранение неясностей** 3](#_Toc118197898)

[**Математическая постановка задачи** 3](#_Toc118197899)

[**Контрольный пример** 3](#_Toc118197900)

[**Особенности решения задачи на компьютере** 4](#_Toc118197901)

[**Организация интерфейса пользователя в файле** 4](#_Toc118197902)

[**Организация данных в программе** 4](#_Toc118197903)

[**Организация ввода/вывода** 5](#_Toc118197904)

[**Представление алгоритма решения задачи** 6](#_Toc118197905)

[**Текст программы** 6](#_Toc118197906)

[**Результаты работы программы** 8](#_Toc118197907)

[**Вывод о проделанной работе** 8](#_Toc118197908)

# **Исходная формулировка задачи**

Для последовательности из *n* вещественных значений создать новую последовательность, состоящую из различных элементов исходной

последовательности и входящих в нее в исходном порядке следования.

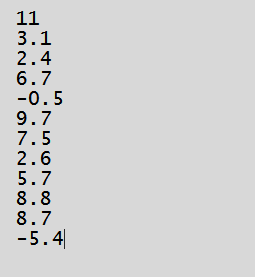
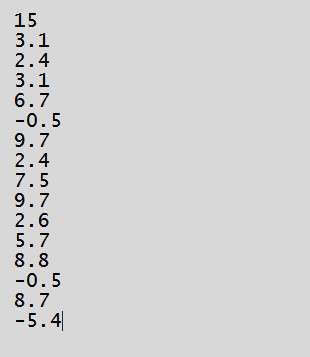
# **Анализ задачи и устранение неясностей**

Программа выполняется в двух вариантах: с использованием одномерного статического и динамического массива. Для статического массива мы задаём константное значение *N*, которая является максимальным размером массива.

# **Математическая постановка задачи**

Необходимо оставить только уникальные значения изначального массива, то есть получить множество значений изначального массива, упорядоченное в исходном порядке следования.

# **Контрольный пример**

****

# Особенности решения задачи на компьютере

При решении задачи использовались переменные типа float, int, unsigned и const unsigned, bool. Отсюда, ], .

# Организация интерфейса пользователя в файле

Макет O1: Задача: Для последовательности из n вещественных значений создать новую последовательность, состоящую из различных элементов исходной последовательности и входящих в нее в исходном порядке следования.

Автор: Коняев Александр Евгеньевич; Группа: 2302; Версия 4.\_

Дата начала: 26.10.2022; Дата окончания:

Макет I1: L

Макет I2: ±d.ddd

Макет O2: Err open\_in

Макет O3: Eof

Макет O4: L

±d.ddd

…

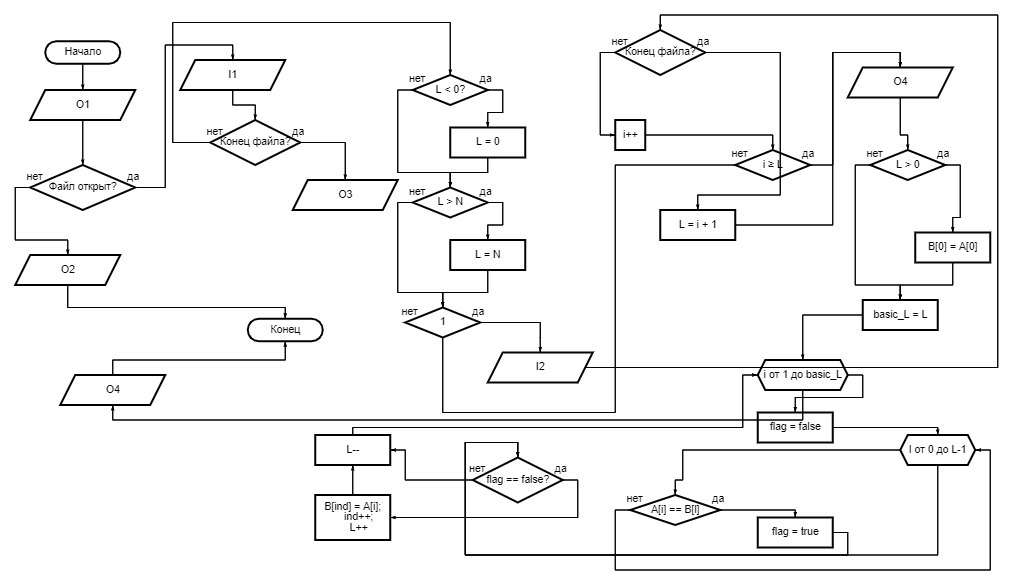
# Организация данных в программе

|  |  |  |
| --- | --- | --- |
| Тип | Название | Назначение |
| float[] | A | Массив чисел, которые берутся из файла in.txt |
| B | Конечный массив чисел |
| int | L | Количество обрабатываемых чисел |
| basic\_L | Переменная, сохраняющая значение L перед заходом в обрабатывающий цикл |
| unsigned | ind | Итерируемая переменная для записи индекса массива |
| i | Итерируемая переменная цикла |
| const unsigned | N | Максимальный размер исходного массива |
| bool | flag | Флаг для проверки условия |
| fstream | f | Файл in.txt |
| g | Файл out.txt |

# Организация ввода/вывода

|  |  |  |
| --- | --- | --- |
|  | iomanip | fstream |
| Ввод |  | f >> |
| Вывод |  | f << |
| Настройка | endl |  |
| Работа с файлом |  | fstream f  f.open()  f.close() |

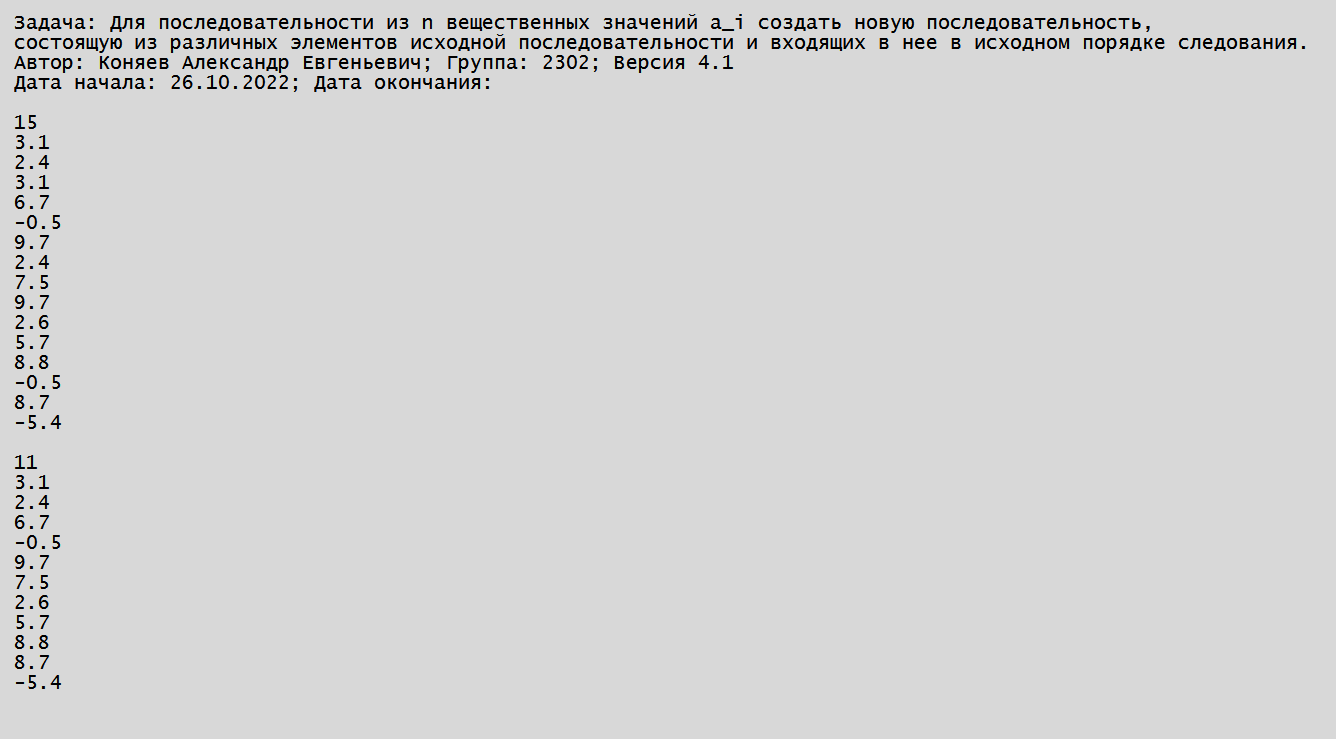
# Представление алгоритма решения задачи



# **Текст программы**

|  |  |
| --- | --- |
| Вариант 1 | Вариант 2 |
| #include <fstream>  #include <iomanip>  int main()  {  const unsigned N = 100;  float A[N], B[N]; int L;  unsigned ind = 1;  std::fstream f, g; f.open("in.txt", std::ios::in); g.open("out.txt", std::ios::out);  g << "Задача: Для последовательности из n вещественных значений a\_i создать новую последовательность,\n"  << "состоящую из различных элементов исходной последовательности и входящих в нее в исходном порядке следования.\n"  << "Автор: Коняев Александр Евгеньевич; Группа: 2302; Версия 4.1\n"  << "Дата начала: 26.10.2022; Дата окончания:\n\n";  if (f.is\_open() == false) g << "Err open\_in";  else {  f >> L; if (f.eof()) g << "Eof";  else {  if (0 > L) L = 0;  else if (L > N) L = N;  unsigned i = 0; while (1) {  f >> A[i];  if (f.eof()) { L = i+1; break; }  else { i++; if (i >= L) break; }  }  }  }  f.close();  g << L << std::endl; for (unsigned i = 0; i < L; i++) g << A[i] << std::endl; g << std::endl;  if (L > 0) B[0] = A[0];  int basic\_L = L;  for (unsigned i = 1; i < basic\_L; i++) {  bool flag = false;  for (unsigned l = 0; l != L - 1; ++l) {  if (A[i] == B[l]) {  flag = true;  break;  }  }  if (flag == false) {  B[ind] = A[i];  ind++;  L++;  }  L--;  }  g << L << std::endl; for (unsigned i = 0; i < L; i++) g << B[i] << std::endl;  g.close();  } | #include <fstream>  #include <iomanip>  int main()  {  int L;  float tmp;  unsigned ind = 1;  std::fstream f, g; f.open("in2.txt", std::ios::in); g.open("out.txt", std::ios::out);  g << "Задача: Для последовательности из n вещественных значений a\_i создать новую последовательность,\n"  << "состоящую из различных элементов исходной последовательности и входящих в нее в исходном порядке следования.\n"  << "Автор: Коняев Александр Евгеньевич; Группа: 2302; Версия 4.2\n"  << "Дата начала: 26.10.2022; Дата окончания:\n\n";  if (f.is\_open() == false) g << "Err open\_in";  else {  if (f.eof()) g << "Eof";  else {  unsigned i = 0; while (!f.eof()) {  f >> tmp;  i++;  }  L = i;  f.close();  }  }  float\* pA = new float[L + 1];  f.open("in2.txt", std::ios::in);  if (f.is\_open() == false) g << "Err open\_in";  else {  for (unsigned i = 0; i < L; ++i) {  f >> \*(pA + i);  }  }  f.close();  g << L << std::endl; for (unsigned i = 0; i < L; i++) g << \*(pA + i) << std::endl; g << std::endl;  int basic\_L = L;  float \*pB = new float[L];  if (L > 0) \*(pB + 0) = \*(pA + 0);  for (unsigned i = 1; i < basic\_L; i++) {  bool flag = false;  for (unsigned l = 0; l != basic\_L - 1; ++l) {  if (\*(pA + i) == \*(pB + l)) {  flag = true;  break;  }  }  if (flag == false) {  \*(pB + ind) = \*(pA + i);  ind++;  L++;  }  L--;  }  g << L << std::endl; for (unsigned i = 0; i < L; i++) g << \*(pB + i) << std::endl;  g.close();  delete[]pA;  delete[]pB;  } |

# **Результаты работы программы**



# **Вывод о проделанной работе**

В ходе выполнения лабораторной работы я научился работать с одномерными статическими и динамическими массивами, указателями и файловым вводом.