ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ПРАКТИЧЕСКОЙ РАБОТЕ №4.1

По дисциплине«Процедурное программирование»

Выполнил: ст. гр.ТКИ – 111

Кох Е.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

**Оглавление**

[1. Условие задачи 3](#_Toc89161491)

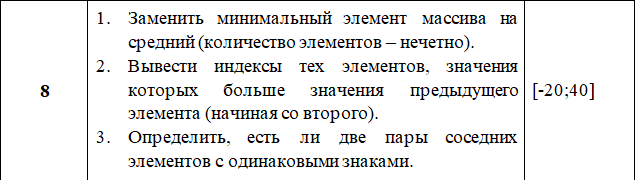
[2. Блок-схема алгоритма 4](#_Toc89161493)

[3. Текст программы на языке C++ 10](#_Toc89161494)

[4. Результаты работы программы 13](#_Toc89161495)

1. Условие задачи

Создать одномерный массив из n целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Составить блок-схему.



1. Блок-схема алгоритма

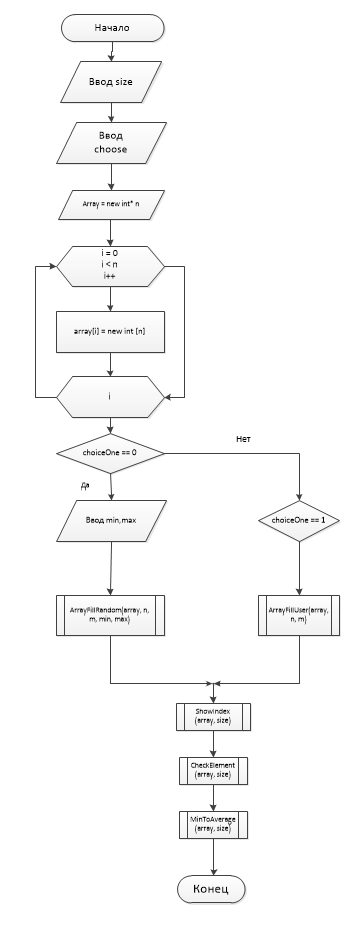


Рисунок1 -Блок-схема main 4.1

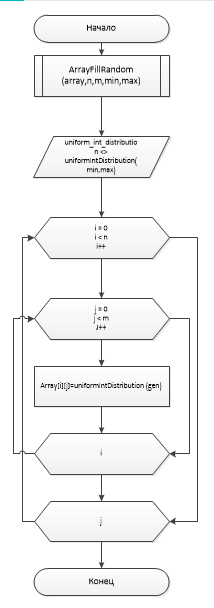


Рисунок – заполнение случайными числами

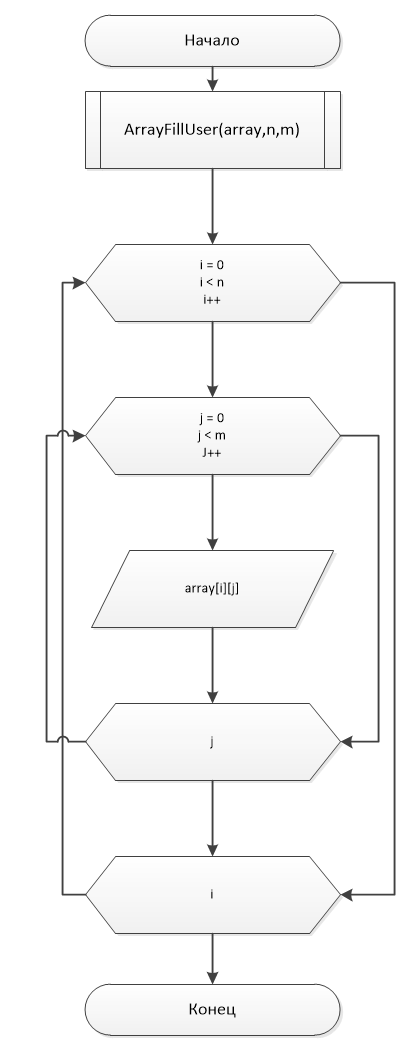


Рисунок – заполнение с клавиатуры

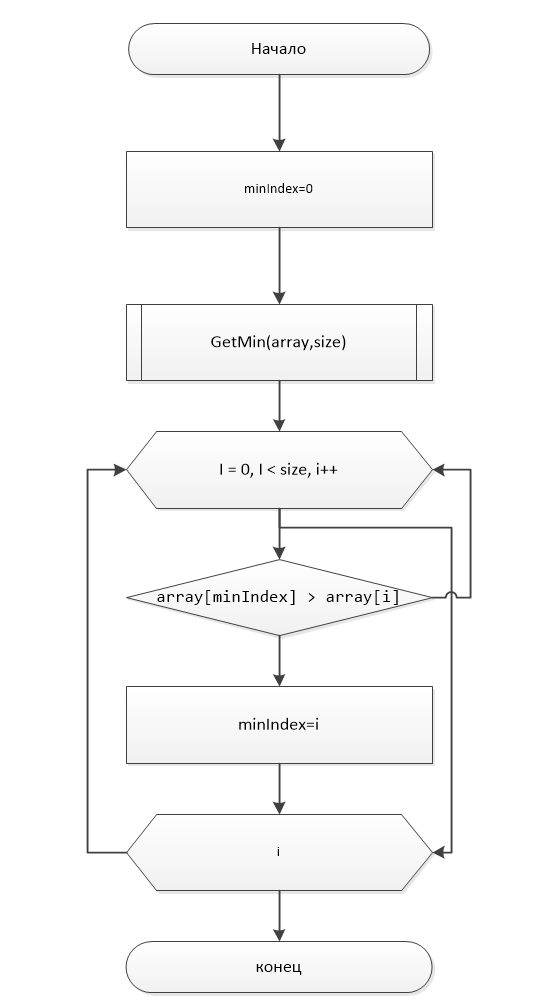


Рисунок – поиск минимального индекса

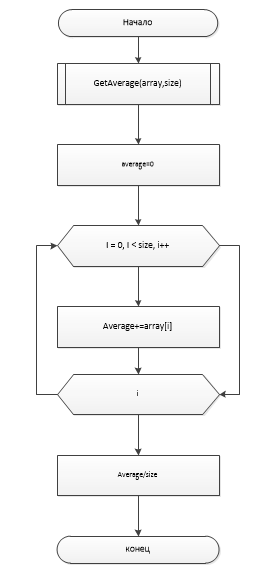


Рисунок – функция нахождение среднего значения

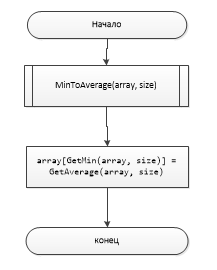


Рисунок – замена минимального на среднее

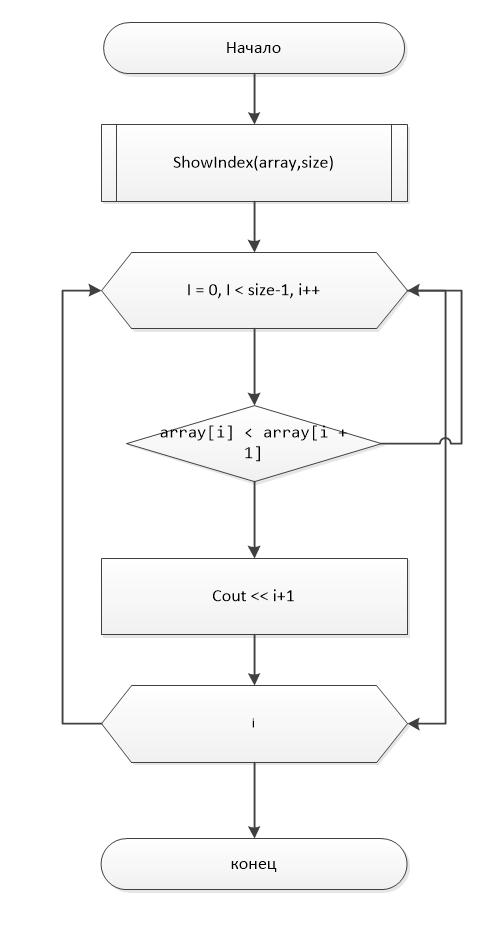


Рисунок – вывод индекса

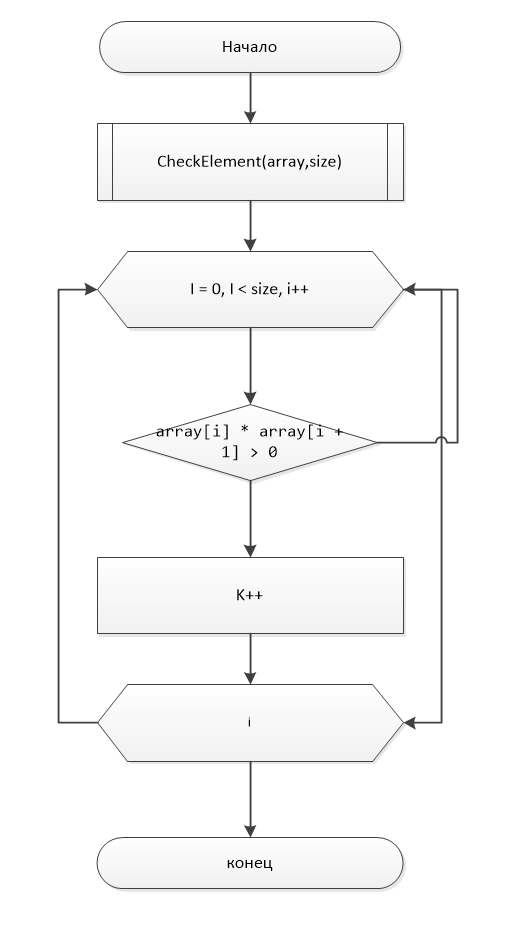


Рисунок – проверка пар

1. Текст программы на языке C++

|  |
| --- |
| #include <iostream>  #include <cmath>  #include <random>  #include <string>  #include <sstream>  #include <Windows.h>  using namespace std;  /\*\*  \*\brief Нахождение минимального по значению элемента  \*\param array Массив  \*\param size Размер массива  \*\*/  size\_t GetMin(const int\* array, const size\_t size);  /\*\*  \*\brief Нахождение среднего арифметического среди элементов массива  \*\param array Массив  \*\param size Размер массива  \*\*/  int GetAverage(const int\* array, const size\_t size);  /\*\*  \*\brief Замена минимального элемента массива на среднее арифметическое  \*\param array Массив  \*\param size Размер массива  \*\*/  void MinToAverage(int\* array, const size\_t size);  /\*\*  \*\brief Вывод индексов элементов, значение которых больше предыдущих  \*\param array Массив  \*\param size Размер массива  \*\*/  void ShowIndex(int\* array, const size\_t size);  /\*\*  \*\brief Проверка на существование двух пар элементов с одинаковыми знаками  \*\param array Массив  \*\param size Размер массива  \*\*/  bool CheckElement(int\* array, const size\_t size);  /\*\*  \* \brief Выбор заполнеия массива  \*\*/  enum class Input  {  randomize = 1,  user = 2  };  /\*\*  \* \brief Заполнение массива случайными числами  \* \array Массив  \* \size Размер массива  \* \param min Минимально возможное число  \* \param max Максимально возможное число  \*\*/  void randomizeArray(int\* array, const size\_t size, const int min, const int max);  /\*\*  \* \brief Ввод массива с клавиатуры  \* \param array Массив  \* \param size Размер массива  \*\*/  void userInput(int\* array, const size\_t size);  /\*\*  \* \brief Конвертация массива строку вида {element\_1, element\_2, ... ,element\_n}  \* \param array Массив  \* \param size Размер массива  \* \param message Сообщение, которое будет выведено перед массивом  \*\*/  string arrayToString(int\* array, const size\_t size, const string& message);  int main() {  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  const int min = -20, max = 40;  cout << "Введите размер массива\n";  size\_t size;  cin >> size;  int\* array = new int[size];  cout << "Выберите способ заполнения массива:\n" << "Случайные элементы - 1\n" << "Ручной ввод - 2\n";  int choice;  cin >> choice;  const auto input = static\_cast<Input>(choice);  switch (input)  {  case Input::randomize:  {  randomizeArray(array, size, min, max);  cout << arrayToString(array, size, "массив = ") << "\n";  break;  }  case Input::user:  {  userInput(array, size);  cout << arrayToString(array, size, "массив = ") << "\n";  break;  }  default:  break;  }  cout << "Индексы элементов, значения которых больше предыдущих\n";  ShowIndex(array, size);  if (CheckElement(array, size))  cout << "существует две пары соседних элементов с одинаковыми знаками\n";  else  cout << "не существует две пары соседних элементов с одинаковыми знаками\n";  MinToAverage(array, size);  cout << arrayToString(array, size, "Измененный массив = ") << '\n';  delete[] array;  }  void randomizeArray(int\* array, const size\_t size, const int min, const int max)  {  std::random\_device rd;  std::mt19937 gen(rd());  const uniform\_int\_distribution<> uniformIntDistribution(min, max);  for (size\_t i = 0; i < size; ++i)  array[i] = uniformIntDistribution(gen);  }  void userInput(int\* array, const size\_t size)  {  for (size\_t i = 0; i < size; i++)  {  cout << "array[" << i << "] = ";  cin >> array[i];  }  }  string arrayToString(int\* array, const size\_t size, const string& message = "")  {  stringstream str;  str << message << "{" << array[0];  for (size\_t i = 1; i < size; ++i)  {  str << ", " << array[i];  }  str << "}";  return str.str();  }  size\_t GetMin(int\* array, size\_t size) {  size\_t minIndex = 0;  for (size\_t i = 1; i < size; i++)  {  if (array[minIndex] > array[i])  minIndex = i;  }  return minIndex;  }  int GetAverage(int\* array, size\_t size) {  int average = 0;  for (size\_t i = 1; i < size; i++) {  average += array[i];  }  return average / size;  }  void MinToAverage(int\* array, size\_t size)  {  array[GetMin(array, size)] = GetAverage(array, size);  }  void ShowIndex(int\* array, size\_t size) {  if (size <= 2)  return;  for (size\_t i = 1; i < size - 1; i++)  if (array[i] < array[i + 1])  cout << "index = " << i + 1 << '\n';  }  bool CheckElement(int\* array, size\_t size) {  int k = 0;  for (size\_t i = 0; i < size; i++)  if (array[i] \* array[i + 1] > 0)  k++;  return k >= 2;  } |

1. Результаты работы программы

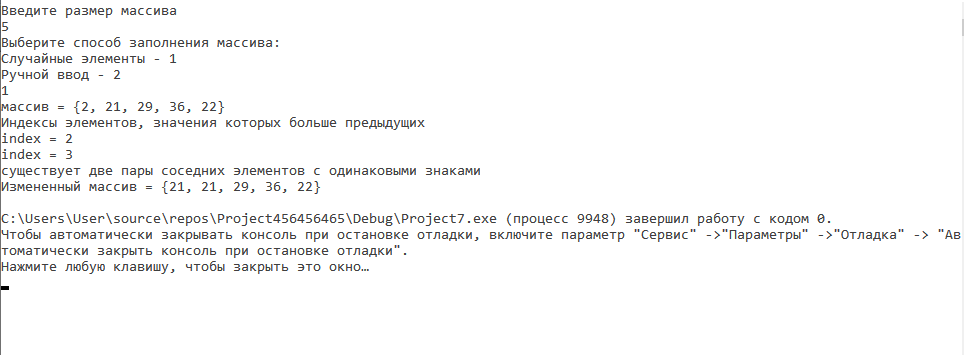


Рисунок 7 -Вывод программы4.1