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

Талатин Д.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

**Оглавление**

[1. Условие задачи 3](#_Toc88682781)

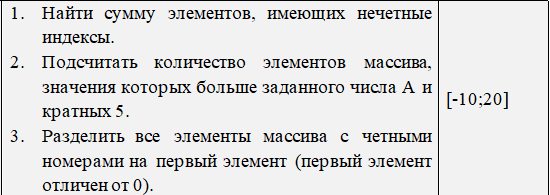
[2. Блок-схема алгоритма 4](#_Toc88682783)

[3. Текст программы на языке C++ 9](#_Toc88682784)

[4. Результаты работы программы 12](#_Toc88682785)

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

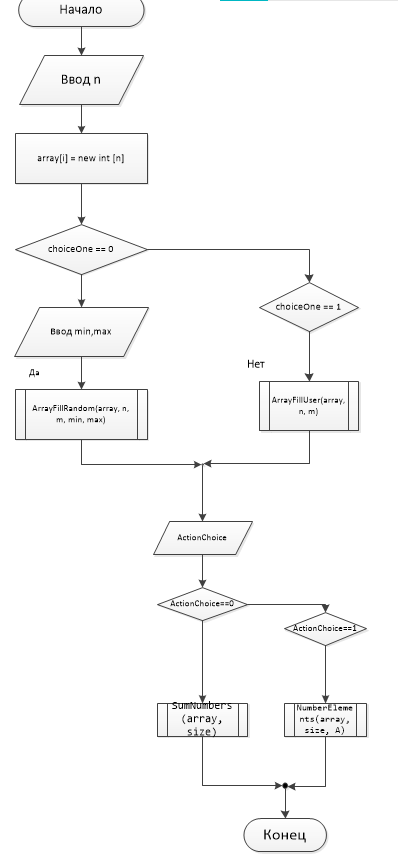


Рисунок1 -Блок-схема main 4.1

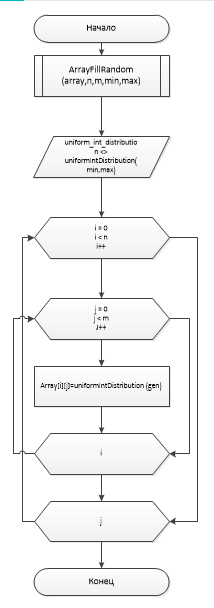


Рисунок – заполнение случайными числами

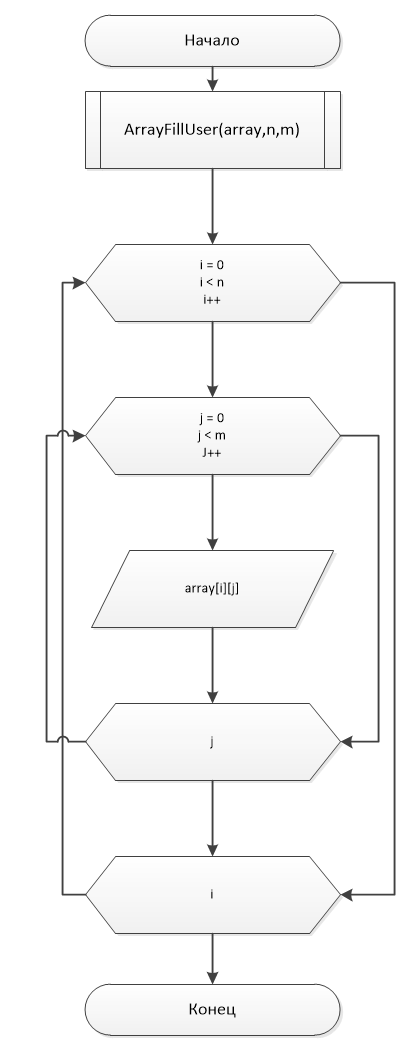


Рисунок – заполнение с клавиатуры

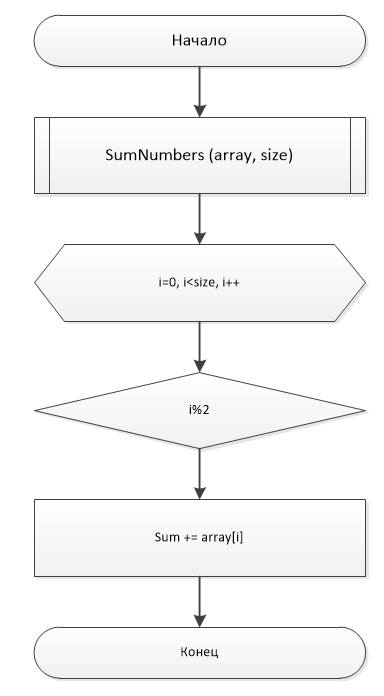


Рисунок – функция подсчета суммы элементов с кратными коэф-и

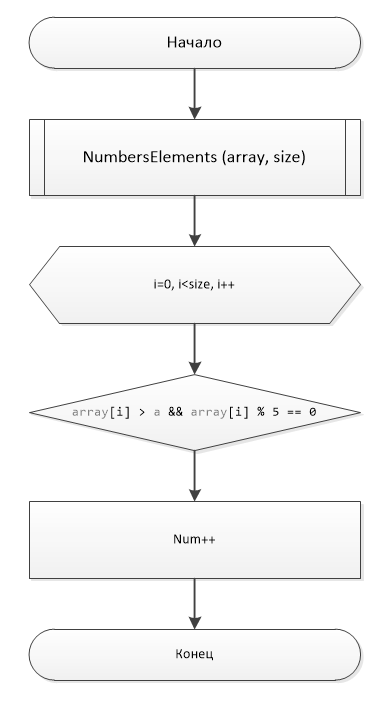


Рисунок – функция подсчета количества элементов

1. Текст программы на языке C++

|  |
| --- |
| #define \_USE\_MATH\_DEFINES  #include <iostream>  #include <cmath>  #include <random>  #include <iomanip>  using namespace std;  /\*\*  \* \brief Заполнение массива случайными числами  \* \param array Массив  \* \param size Размер массива  \* \param min Минимально возможное число  \* \param max Максимально возможное число  \*\*/  void RandomArray(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 UserArray(int\* array, const size\_t size);  /\*\*  \* \brief Сумма элементов с нечетными индексами  \* \param array Массив  \* \param size Размер массива  \*\*/  int SumNumbers(int\* array, const size\_t size);  /\*\*  \* \brief Количество элементов(значения которых больше А, кратны 5)  \* \param array Массив  \* \param size Размер массива  \* \param A значение по условию  \*\*/  int NumberElements(int\* array, const size\_t size, int a);  /\*\*  \* \brief Новый массив(все четные элементы делить на первый элемент)  \* \param array Массив  \* \param second\_array Обработанный массив  \* \param size Размер массива  \*\*/  int SecondArray(int\* array, double\* second\_array, const size\_t size);  enum class ArrayType  {  RandomNumber = 1,  UserNumber = 2  };  int main()  {  cout.precision(3);  cout << fixed;  const int MIN = -10, MAX = 20;  int sum, A;  cout << "Please input array length: ";  size\_t size;  cin >> size;  cout << "Enter the number A by condition: ";  cin >> A;  int\* array = new int[size];  double\* second\_array = new double[size];  cout << "\nChoose array:\n" << "1) Array with random number\n" << "2) Array with user number\n" << "1 or 2: ";  int choice;  cin >> choice;  const auto arraytype = static\_cast<ArrayType>(choice);  switch (arraytype)  {  case ArrayType::RandomNumber:  {  RandomArray(array, size, MIN, MAX);  break;  }  case ArrayType::UserNumber:  {  UserArray(array, size);  break;  }  default:  break;  }  cout << SumNumbers(array, size) << endl;  cout << NumberElements(array, size, A) << endl;  int secondArray = SecondArray(array, second\_array, size);  for (size\_t i = 0; i < size; i++)  {  cout << "array[" << i << "] = " << second\_array[i] << endl;  }  delete[] array;  delete second\_array;  }  void RandomArray(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);  for (size\_t i = 0; i < size; ++i)  {  cout << "array[" << i << "] = " << array[i] << endl;  }  }  void UserArray(int\* array, const size\_t size)  {  for (size\_t i = 0; i < size; i++)  {  cout << "array[" << i << "] = ";  cin >> array[i];  }  }  int SumNumbers(int\* array, const size\_t size)  {  int sum = 0;  for (size\_t i = 0; i < size; ++i)  {  if (i % 2)  {  sum += array[i];  }  }  return sum;  }  int NumberElements(int\* array, const size\_t size, int a)  {  int num = 0;  for (size\_t i = 0; i < size; ++i)  {  if (array[i] > a && array[i] % 5 == 0)  {  num++;  }  }  return num;  }  int SecondArray(int\* array, double\* second\_array, const size\_t size)  {  double first = array[0];  for (size\_t i = 0; i < size; i++)  {  if (i % 2)  {  second\_array[i] = array[i] / first;  }  else {  second\_array[i] = array[i];  }  }  return first;  } |

1. Результаты работы программы

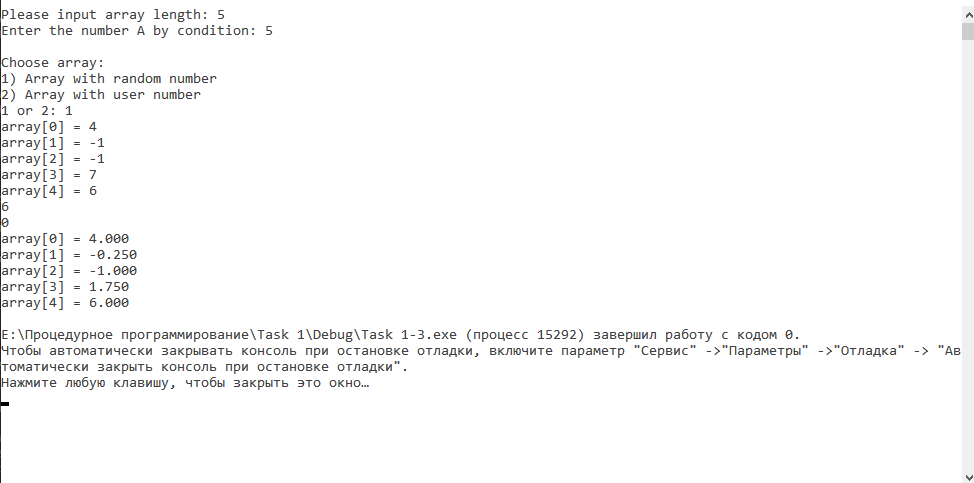


Рисунок 6 -Вывод программы4.1