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

По дисциплине «Введение в языки программирования»

Выполнил: ст. гр. ТКИ – 111

Соколов И.Д.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2022

Оглавление

[Задание 4–1 3](#_Toc85287969)

[Формулировка задания 3](#_Toc85287970)

[Блок-схема алгоритма 4](#_Toc85287971)

[Программа на языке С++ 5](#_Toc85287972)

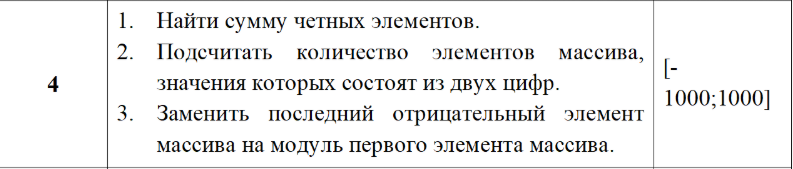
[Решение тестовых примеров на С++ 9](#_Toc85287973)

**Задание 4-1**

**Формулировка задания**

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.

**Таблица 1**



**Блок-схемаалгоритма**

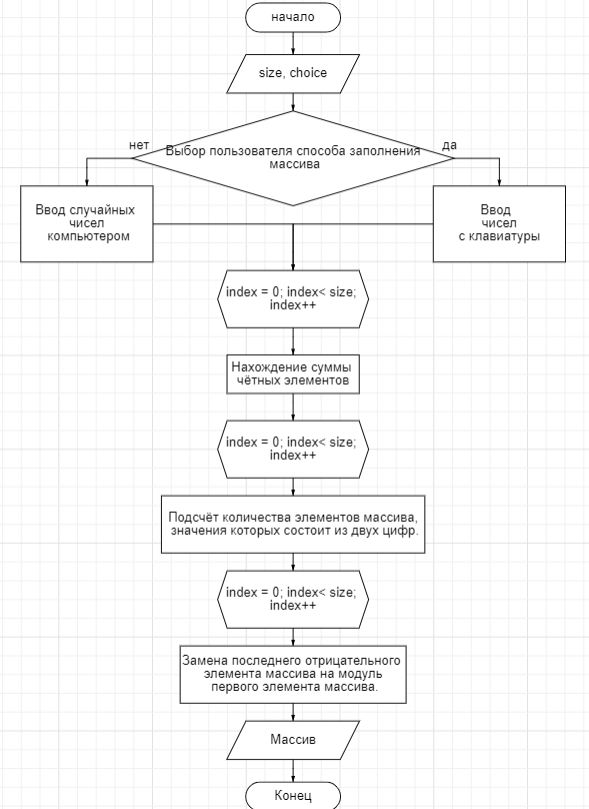
****

Рисунок 1 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |
| --- | --- |
|  | #include <iostream> |
|  | #include <random> |
|  | #include <cmath> |
|  |  |
|  | using namespace std; |
|  |  |
|  | /\*\* |
|  | \* \brief Проверка ввода размера массива. |
|  | \* \return Размер массива. |
|  | \*/ |
|  | size\_t GetSize(); |
|  |  |
|  | /\*\* |
|  | \* \brief Подсчёт цифр в числе. |
|  | \* \param number Число. |
|  | \* \return Количество цифр. |
|  | \*/ |
|  | int CountNumbers(int number); |
|  |  |
|  | /\*\* |
|  | \* \brief Замена последнего отрицательного элемента массива на модуль первого элемента. |
|  | \* \param myArray массив. |
|  | \* \param size размер массива. |
|  | \* \return массив. |
|  | \*/ |
|  | int\* ChangeArrayElements( int\* myArray, const size\_t size); |
|  |  |
|  | /\*\* |
|  | \* \brief Заполнение массива случайными числами. |
|  | \* \param size размер массива. |
|  | \* \param minValue минимальное значение элементов массива. |
|  | \* \param maxValue максимальное значение элементов массива. |
|  | \* \return заполненный массив. |
|  | \*/ |
|  | int\* FillRandomArray(size\_t size, int minValue, int maxValue); |
|  |  |
|  | /\*\* |
|  | \* \brief Вывод массива на консоль. |
|  | \* \param myArray массив. |
|  | \* \param size размер массива. |
|  | \*/ |
|  | void ArrayPrint(const int\* myArray, const size\_t size); |
|  |  |
|  | /\*\* |
|  | \* \brief Функция считает количество элементов массива, значения которых состоят из двух цифр. |
|  | \* \param myArray массив. |
|  | \* \param size размер массива. |
|  | \* \return количество элементов. |
|  | \*/ |
|  | int CountElements(const int\* myArray, const size\_t size); |
|  |  |
|  | /\*\* |
|  | \* \brief Вычисление суммы четных элементов массива. |
|  | \* \param myArray массив. |
|  | \* \param size размер массива. |
|  | \* \return Сумма. |
|  | \*/ |
|  | int SumOfEvenElements(const int\* myArray, const size\_t size); |
|  |  |
|  | /\*\* |
|  | \* \brief Метод, возвращающий заполненный пользователем массив. |
|  | \* \param size размер массива. |
|  | \* \param minValue минимальное значение элементов массива. |
|  | \* \param maxValue максимальное значение элементов массива. |
|  | \* \return заполненный массив. |
|  | \*/ |
|  | int\* FillUserArray(size\_t size); |
|  |  |
|  | /\*\* |
|  | \* \brief Варианы ввода массива. |
|  | \*/ |
|  | enum class ArrayInputWay |
|  | { |
|  | random, |
|  | keyboard |
|  | }; |
|  |  |
|  | /\*\* |
|  | \* \brief Точка входа в программу. |
|  | \* \return 0, в случае успеха. |
|  | \*/ |
|  | int main() |
|  | { |
|  | setlocale(LC\_ALL, "Russian"); |
|  | size\_t size = GetSize(); |
|  |  |
|  | if (size == 0) |
|  | return 1; |
|  |  |
|  | cout << "Как вы хотите заполнить массив?\n"; |
|  | cout << static\_cast<int>(ArrayInputWay::random) << " - random,\n"; |
|  | cout << static\_cast<int>(ArrayInputWay::keyboard) << " - keyboard.\n"; |
|  | cout << "Ваш выбор: "; |
|  | int choice; |
|  | cin >> choice; |
|  |  |
|  | const auto chosen = static\_cast<ArrayInputWay>(choice); |
|  | int\* myArray = nullptr; |
|  |  |
|  | auto minValue = 0; |
|  | auto maxValue = 0; |
|  | cout << "Введите диапазон чисел массива (сначала минимум, потом максимум) " << endl; |
|  | cin >> minValue >> maxValue; |
|  | if (maxValue <= minValue) |
|  | { |
|  | cout << "Введен неправильный диапазон!" << endl; |
|  | } |
|  |  |
|  | switch (chosen) |
|  | { |
|  | case ArrayInputWay::random: |
|  | { |
|  | myArray = FillRandomArray(size, minValue, maxValue); |
|  | break; |
|  | } |
|  | case ArrayInputWay::keyboard: |
|  | { |
|  | myArray = FillUserArray(size); |
|  | break; |
|  | } |
|  | } |
|  |  |
|  | ArrayPrint(myArray, size); |
|  |  |
|  | cout << "Сумма чётных элементов массива: " << SumOfEvenElements(myArray, size) << endl; |
|  |  |
|  | cout << "Количество элементов, состоящих из 2х цифр: " << CountElements(myArray, size) << endl; |
|  |  |
|  | cout << "Массив с заменёнными элементами: "; |
|  | myArray = ChangeArrayElements(myArray, size); |
|  | ArrayPrint(myArray, size); |
|  |  |
|  | if (myArray != nullptr) { |
|  |  |
|  | delete[] myArray; |
|  | myArray = nullptr; |
|  |  |
|  | } |
|  | return 0; |
|  |  |
|  | } |
|  |  |
|  | size\_t GetSize() { |
|  | int size = 0; |
|  | cout << "Введите размер массива" << endl; |
|  | cin >> size; |
|  | if (size <= 0) |
|  | { |
|  | cout << "Введён неверный размер"; |
|  | return 0; |
|  | } |
|  | else |
|  | return size; |
|  | }; |
|  |  |
|  | int\* ChangeArrayElements(int\* myArray, const size\_t size) { |
|  |  |
|  | for (int index = size; index >= 0; index = index - 1) { |
|  | if (myArray[index] < 0) { |
|  | myArray[index] = abs(myArray[0]); |
|  | break; |
|  | } |
|  | } |
|  | return myArray; |
|  | } |
|  |  |
|  | int CountElements(const int\* myArray, const size\_t size) { |
|  | int countelements = 0; |
|  |  |
|  | for (size\_t index = 0; index < size; index++) { |
|  | if (CountNumbers(myArray[index]) == 2) { |
|  | countelements++; |
|  | } |
|  | } |
|  | return countelements; |
|  |  |
|  | } |
|  |  |
|  | int CountNumbers(int number) { |
|  | int countnumbers = 0; |
|  | while (number > 0){ |
|  | countnumbers++; |
|  | number = number / 10; |
|  | } |
|  | return countnumbers; |
|  | } |
|  |  |
|  | int SumOfEvenElements(const int\* myArray, const size\_t size) |
|  | { |
|  | int sum = 0; |
|  | for (size\_t index = 0; index < size; index++) { |
|  | if (myArray[index] % 2 == 0) |
|  | sum = sum + myArray[index]; |
|  | } |
|  | return sum; |
|  | } |
|  |  |
|  | void ArrayPrint(const int\* myArray, const size\_t size) |
|  | { |
|  | if (myArray == nullptr) |
|  | { |
|  | cout << "Массива не существует"; |
|  | } |
|  | else { |
|  | cout << "\nМассив:\n"; |
|  | for (size\_t index = 0; index < size; index++) { |
|  | cout << myArray[index] << " "; |
|  | } |
|  | cout << "\n"; |
|  |  |
|  | } |
|  | } |
|  |  |
|  | int\* FillRandomArray(const size\_t size, const int minValue, const int maxValue) |
|  | { |
|  | random\_device rd; |
|  |  |
|  | mt19937 gen(rd()); |
|  |  |
|  | const std::uniform\_int\_distribution<> uniformIntDistribution(minValue, maxValue); |
|  |  |
|  | auto\* myArray = new int[size]; |
|  |  |
|  | for (size\_t index = 0; index < size; index++) |
|  | { |
|  | myArray[index] = uniformIntDistribution(gen); |
|  | } |
|  | return myArray; |
|  | } |
|  |  |
|  | int\* FillUserArray(const size\_t size) |
|  | { |
|  | auto\* array = new int[size]; |
|  | cout << "Введите элементы массива" << "\n"; |
|  | for (size\_t index = 0; index < size; index++) |
|  | { |
|  | cin >> array[index]; |
|  | } |
|  | return array; |
|  | } |

**Решение тестовых примеров на языке C++**

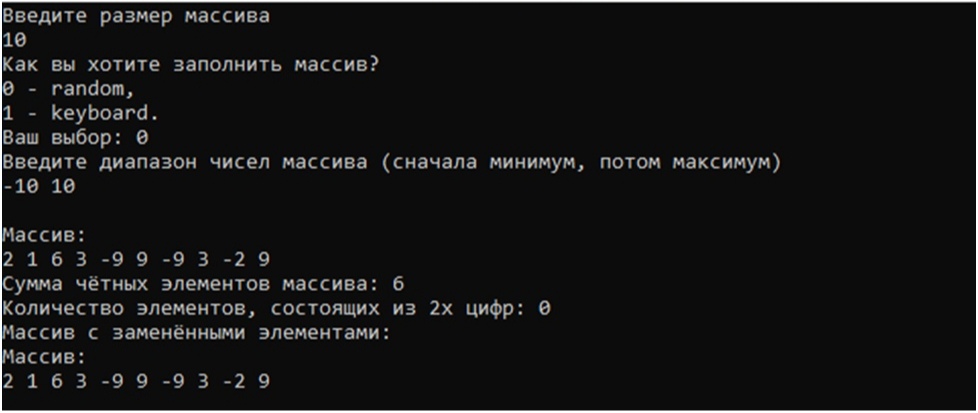


Рисунок 2 – Решение тестовых примеров на C++