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

Самохвалова М. Д.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[**Задание 4–1** 3](#_Toc93852531)

[**Условие задания** 3](#_Toc93852532)

[**Блок-схема алгоритма** 4](#_Toc93852533)

[**Программа на языке С++** 7](#_Toc93852534)

[**Решение тестовых примеров на С++** 11](#_Toc93852535)

# **Задание 4–1**

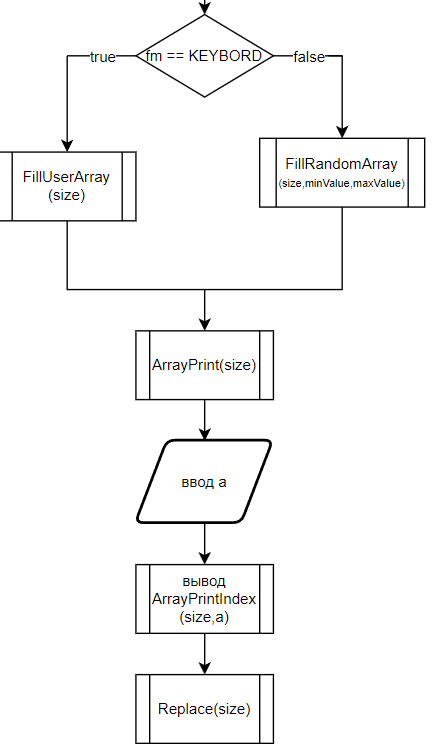
## **Условие задания**

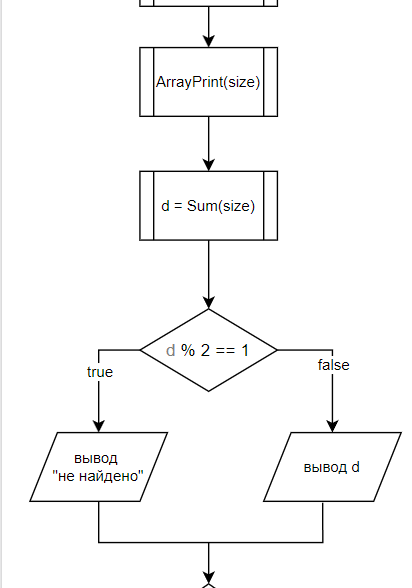
Создать одномерный массив из n целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Составить блок-схему.

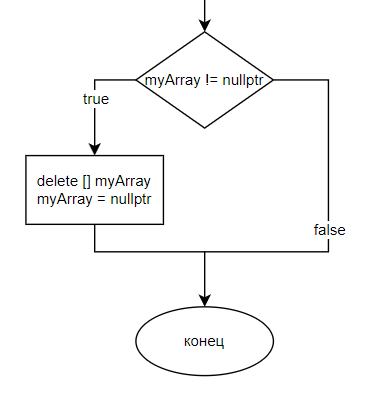
Таблица 1

|  |  |  |
| --- | --- | --- |
| Вариант | Задачи | Интервал |
| 1 | 1. Найти сумму элементов, имеющих нечётное значение. 2. Вывести индексы тех элементов, значения которых больше заданного числа А. 3. Заменить второй элемент массива на максимальный среди отрицательных. | [-10;10] |

## **Блок-схема алгоритма**







## **Программа на языке С++**

#include <iostream>

#include <string>

#include <random>

using namespace std;

/\*\*

\* \brief Проверка ввода размера массива.

\* \return Размер массива.

\*/

size\_t GetSize();

/\*\*

\* \brief Вывод элементов массива, значения которых больше заданного числа А.

\* \param myArray массив.

\* \param size размер массива.

\*/

void ArrayPrintIndex(const 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 для нахождения суммы элементов, имеющих нечетное значение.

\* \param myArray массив.

\* \param size размер массива.

\* \return Сумма элементов, имеющих нечетное значение.

\*/

int Sum(const int\* myArray, const size\_t size);

/\*\*

\* \brief Замена второго элемента массива на максимальный среди отрицательных.

\* \param myArray массив.

\* \param size размер массива.

\* \return maxValue максимальное значение, которое может принимать элемент массива.

\*/

int\* Replace(int\* myArray, const size\_t size, const int minValue);

/\*\*

\* \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()

{

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 нечетных элементов массива: " << Sum(myArray, size) << endl;

Sum(myArray, size);

cout << "Индексы элементов больших а" << endl;

ArrayPrintIndex(myArray, size);

cout << endl;

Replace(myArray, size, minValue);

ArrayPrint(myArray, size);

myArray = Replace(myArray, size, maxValue);

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 Sum(const int\* myArray, const size\_t size) {

int sum = 0;

for (size\_t index = 0; index < size; index++) {

if (myArray[index] % 2 == 1)

sum += myArray[index]; {

}

}

return sum;

}

void ArrayPrintIndex(const int\* myArray, const size\_t size) {

int a;

cout << "Введите a" << endl;

cin >> a;

for (size\_t index = 0; index < size; index++) {

if ((myArray[index]) > a) {

cout << index << " ";

}

}

}

int\* Replace(int\* myArray, const size\_t size, const int minValue)

{

int temprary = 0;

if (myArray == nullptr)

cout << "Массив пуст";

auto maxNegativeArrayValue = minValue;

size\_t maxNegativeElementIndex = 0;

for (size\_t index = 0; index < size; index++) {

if (myArray[index] <0 && myArray[index] > maxNegativeArrayValue) {

maxNegativeArrayValue = myArray[index];

maxNegativeElementIndex = index;

}

}

if (maxNegativeArrayValue > 0) {

cout << "Ошибка";

}

else {

temprary = myArray[maxNegativeElementIndex];

myArray[maxNegativeElementIndex] = myArray[(size - 1) / 2];

myArray[(size - 1) / 2] = temprary;

}

return myArray;

}

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 С++**

