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

Родин К. П.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[**Задание 4–1** 3](#_Toc93875959)

[**Условие задания** 3](#_Toc93875960)

[**Блок-схема алгоритма** 4](#_Toc93875961)

[**Программа на языке С++** 7](#_Toc93875962)

[**Решение тестовых примеров на С++** 11](#_Toc93875963)

[**Задание 4–2** 12](#_Toc93875964)

[**Условие задания** 12](#_Toc93875965)

[**Блок-схема алгоритма** 13](#_Toc93875966)

[**Программа на языке С++** 16](#_Toc93875967)

[**Решение тестовых примеров на С++** 20](#_Toc93875968)

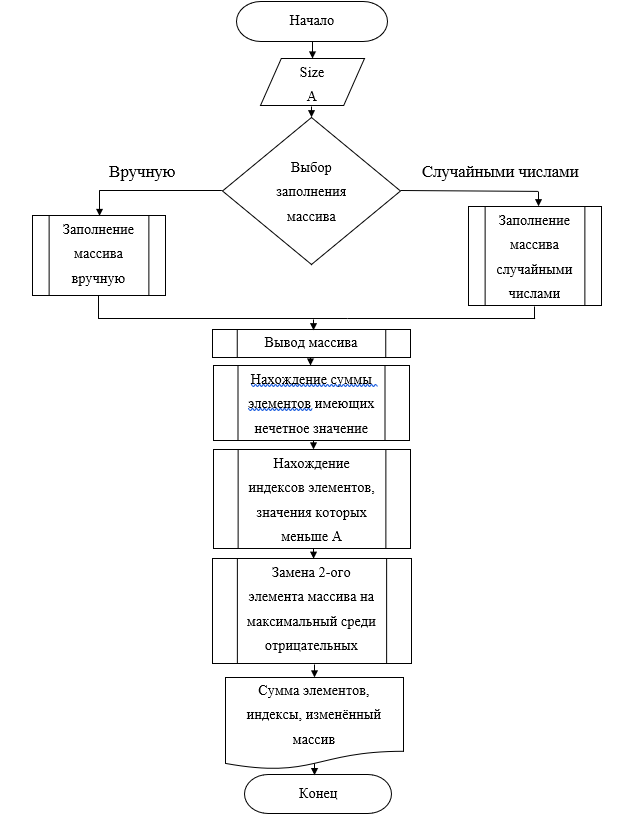
# **Задание 4–1**

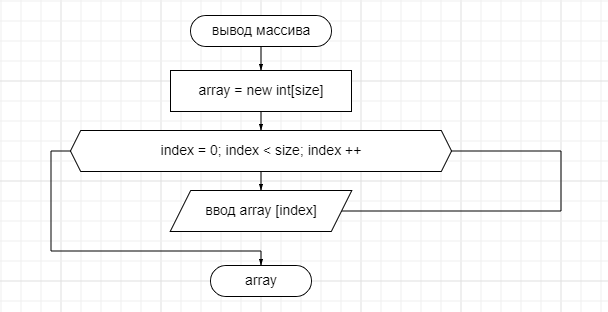
## **Условие задания**

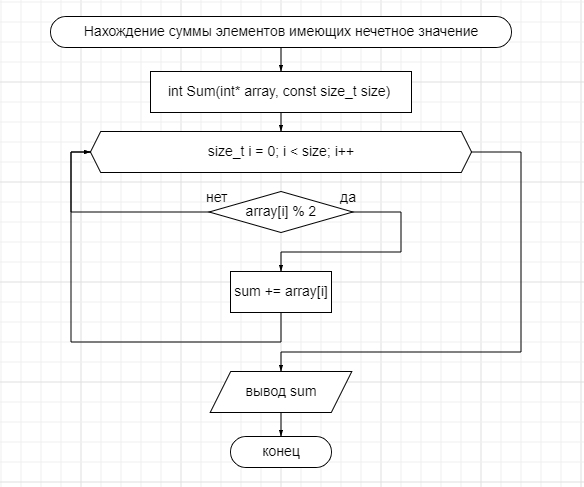
Создать одномерный массив из *n* целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Составить блок-схему.

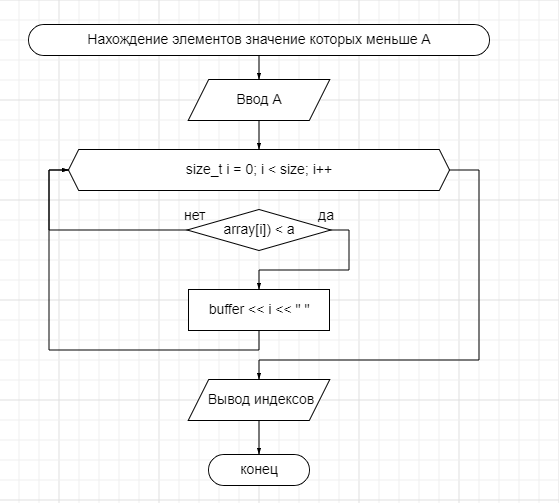
|  |  |  |
| --- | --- | --- |
| Вариант | Задачи | Интервал |
| 1 | 1. Найти сумму элементов, имеющих нечетное значение. 2. Вывести индексы тех элементов, значения которых больше заданного числа А. 3. Заменить второй элемент массива на максимальный среди отрицательных. | [-10;10] |

## **Блок-схема алгоритма**









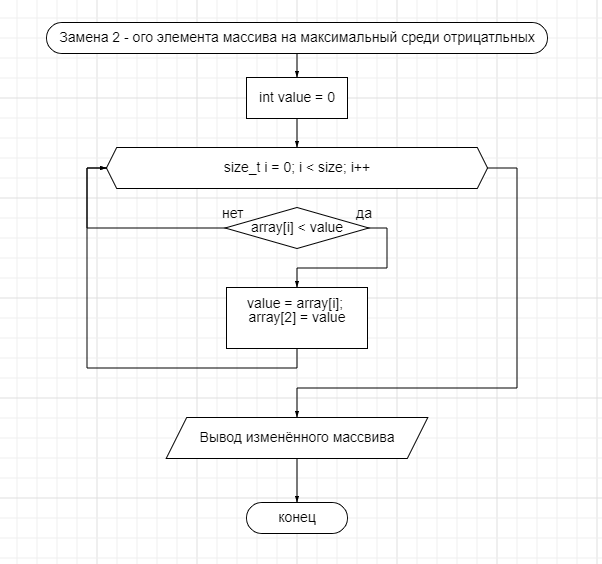


Рисунок 1 – Блок-схема 4-1

## **Программа на языке С++**

#include <iostream>

#include <sstream>

#include <random>

using namespace std;

/\*\*

\* \brief Способы задания массива

\*/

enum class Choice

{

/\*\*

\* \brief Ввод вручную

\*/

Manual,

/\*\*

\* \brief Ввод с помощью рандномных чисел

\*/

Random

};

/\*\*

\* \brief Метод заполнения массива

\* \param size Размер массива

\* \param selection Выбор создания массива (вручную или случайными числами)

\* \param min\_value Минимальное значение в интервале (-10)

\* \param maxv\_alue Максимальное значение в интервале (10)

\* \return Массив

\*/

int\* GetArray(const size\_t size, const int selection, const int min\_value, const int max\_value);

/\*\*

\* \brief Получение массива

\* \param message Мотивоционное сообщение для пользователя

\* \return Размер массива

\*/

size\_t GetSize(const string& message);

/\*\*

\* \brief Вывод элементов массива

\* \param size Размер массива

\* \return Строка со значениями индексов массива

\*/

string ToString(const int\*, const size\_t size);

/\*\*

\* \brief Функция для нахождения суммы элементов, имеющих нечетное значение

\* \param size Размер массива

\* \return Сумма элементов, имеющих нечетное значение

\*/

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

/\*\*

\* \brief Вывод индексов элементов массива, значение которых меньше А

\* \param size Размер массива

\* \param a Значение А, вводимое пользователем

\* \return Строка с индексами элементов массива, значения которых меньше А

\*/

string ToStringIndex(const int\*, const size\_t size, const int a);

/\*\*

\* \brief Функция для замены второго элемента массива на максимальный (математически) среди отрицательных

\* \param size Размер массива

\*/

void Replace(int\*, const size\_t size);

/\*\*

\* \brief Точка входа в программу

\* \return (Код ошибки 0) успех

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

auto error\_code = 0;

int\* my\_array = nullptr;

const int min\_value = -10;

const int max\_value = 10;

try

{

const auto size = GetSize("Введите размер массива: ");

cout << "Выберите способ создания массива: " << static\_cast<int>(Choice::Manual) << " - вручную, " << static\_cast<int>(Choice::Random) << " - заполнить случайными числами ";

int input\_type;

cin >> input\_type;

my\_array = GetArray(size, input\_type, min\_value, max\_value);

cout << "Итоговый массив:\n";

cout << ToString(my\_array, size);

cout << "\nСумма элементов, имеющих нечетное значение: " << Sum(my\_array, size) << '\n';

int a;

cout << "Введите число A: ";

cin >> a;

cout << "Индексы: ";

cout << ToStringIndex(my\_array, size, a);

cout << "\nМассив после замены второго элемента массива на максимальный среди отрицательных:\n";

Replace(my\_array, size);

cout << ToString(my\_array, size);

}

catch (exception& e)

{

cout << e.what();

error\_code = 1;

}

if (my\_array != nullptr)

{

delete[] my\_array;

my\_array = nullptr;

}

return error\_code;

}

size\_t GetSize(const string& message)

{

int size = -1;

cout << message;

cin >> size;

if (size <= 0)

{

throw out\_of\_range("Неправильный размер массива");

}

return size;

}

int\* GetArray(const size\_t size, const int selection, const int min\_value, const int max\_value)

{

const auto array = new int[size];

//Will be used to obtain a seed for the random number engine

random\_device rd;

//Standard mersenne\_twister\_engine seeded with rd()

mt19937 gen(rd());

const uniform\_int\_distribution<> uniformIntDistribution(min\_value, max\_value);

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

{

switch (selection)

{

case static\_cast<int>(Choice::Manual):

{

cout << "Введите " << index + 1 << " элемент массива в диапозоне [-10;10]: ";

cin >> array[index];

break;

}

case static\_cast<int>(Choice::Random):

{

array[index] = uniformIntDistribution(gen);

break;

}

}

}

return array;

}

string ToString(const int\* array, const size\_t size)

{

if (array == nullptr)

throw invalid\_argument("Массив не существует");

stringstream buffer;

buffer << "{";

for (size\_t index = 0; index < size - 1; index++)

{

buffer << array[index] << ", ";

}

buffer << array[size - 1] << "}";

return buffer.str();

}

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

int sum = 0;

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

if (array[i] % 2)

sum += array[i];

}

return sum;

}

string ToStringIndex(const int\* array, const size\_t size, const int a)

{

stringstream buffer;

for (size\_t i = 0; i < size; i++)

{

if ((array[i]) < a)

{

buffer << i << " ";

}

}

return buffer.str();

}

void Replace(int\* array, const size\_t size) {

int value = 0;

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

if (array[i] < value) {

value = array[i];

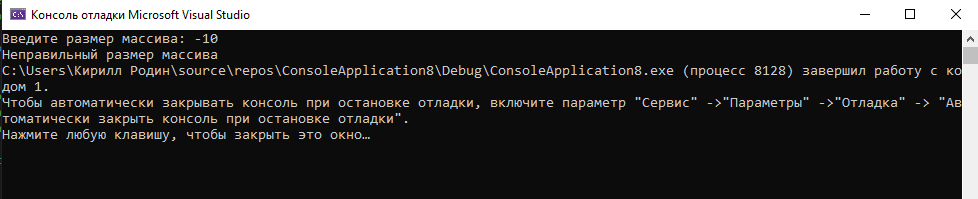
array[2] = value;

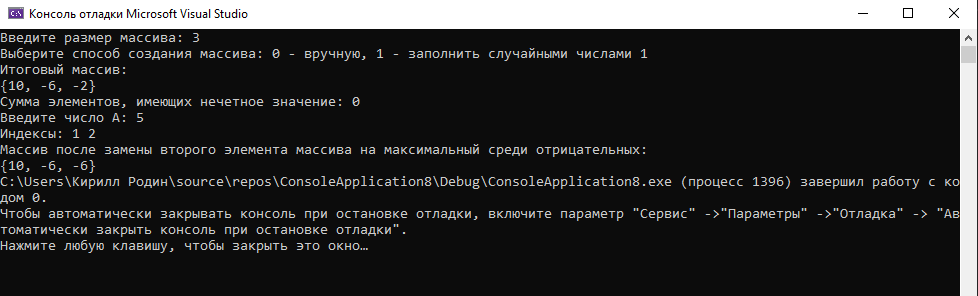
}

}

}

## **Решение тестовых примеров на С++**





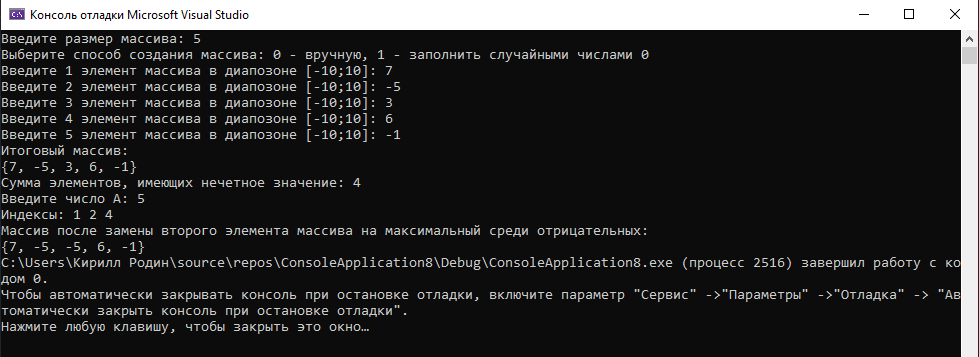


Рисунок 2 - Решение тестовых примеров для задания 4-1

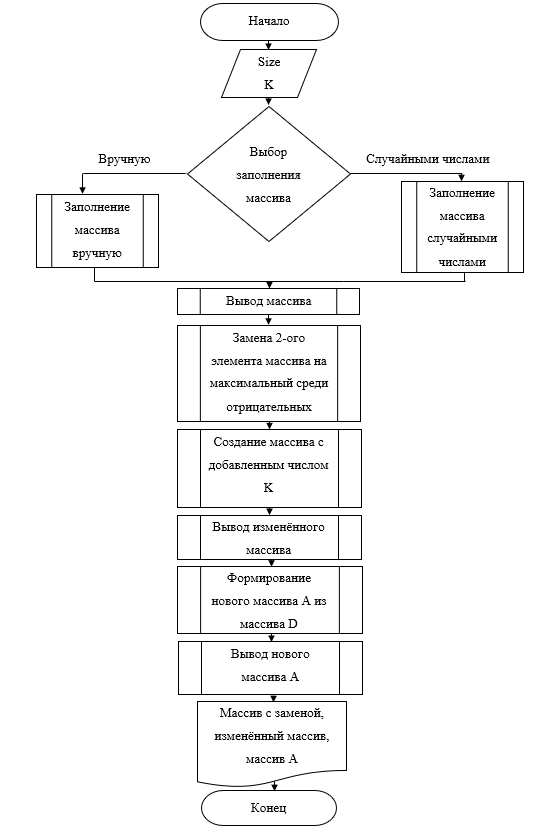
# **Задание 4–2**

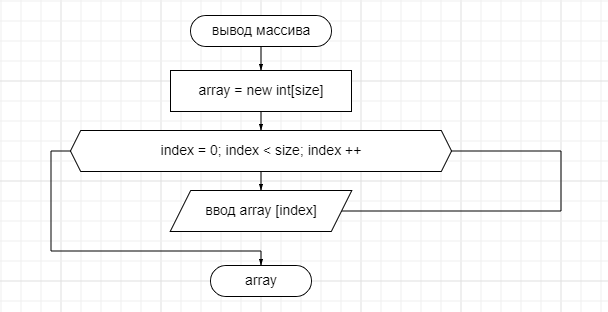
## **Условие задания**

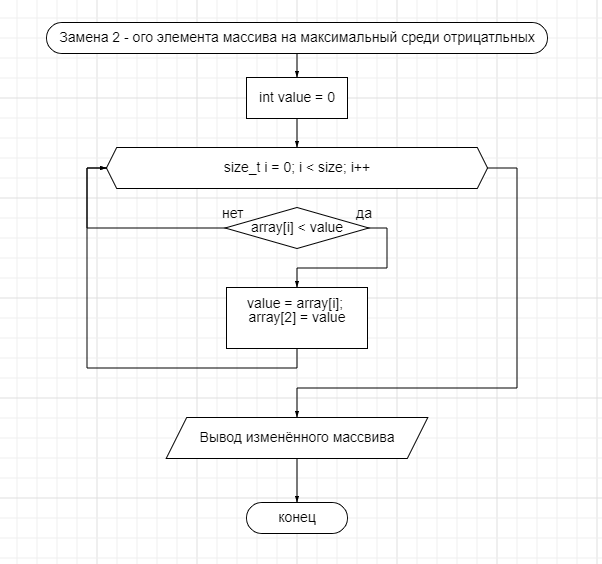
Создать одномерный массив из *n* целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Вывести массив на экран. Составить блок-схему.

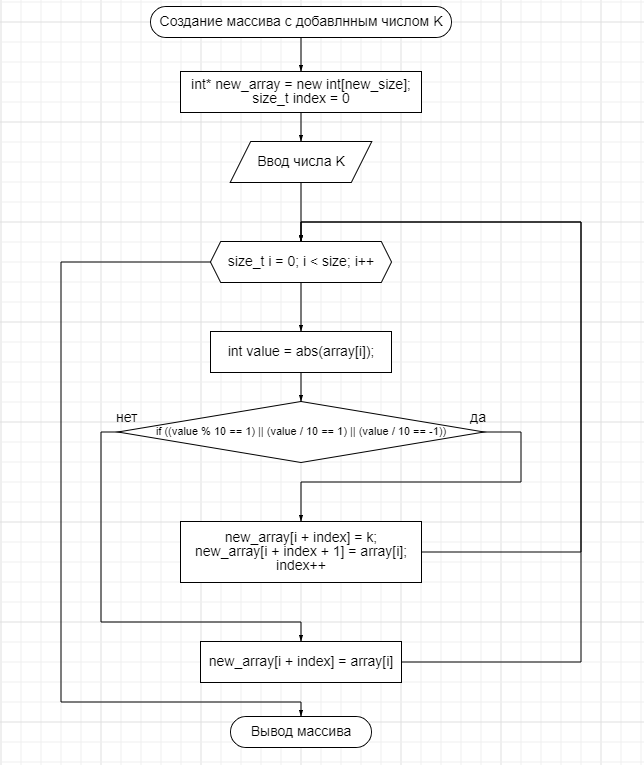
|  |  |  |
| --- | --- | --- |
| Вариант | Задачи | Интервал |
| 1 | 1. Заменить второй элемент массива на максимальный среди отрицательных. 2. Вставить число К перед всеми элементами, в которых есть цифра 1. 3. Из элементов массива D сформировать массив A той же размерности по правилу: если номер четный, то значение элемента находится по формуле Ai = Di2, а если нечетный, то по формуле Ai = Di / i. | [-10;10] |

## **Блок-схема алгоритма**









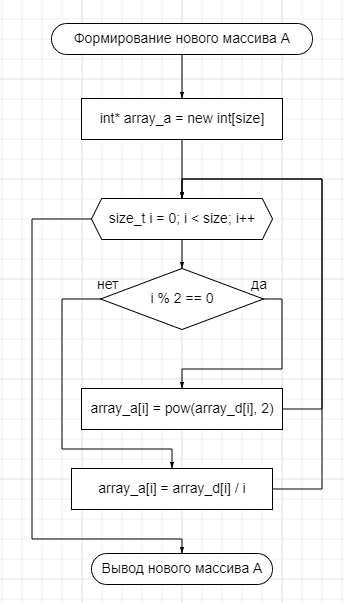


Рисунок 3 - Блок-схема 4-2

## **Программа на языке С++**

#include <iostream>

#include <sstream>

#include <random>

using namespace std;

/\*\*

\* \brief Способы задания массива

\*/

enum class Choice

{

/\*\*

\* \brief Ввод вручную

\*/

Manual,

/\*\*

\* \brief Ввод с помощью рандномных чисел

\*/

Random

};

/\*\*

\* \brief Метод заполнения массива

\* \param size Размер массива

\* \param selection Выбор создания массива (вручную или случайными числами)

\* \param min\_value Минимальное значение в интервале (-10)

\* \param maxv\_alue Максимальное значение в интервале (10)

\* \return Массив

\*/

int\* GetArray(const size\_t size, const int selection, const int min\_value, const int max\_value);

/\*\*

\* \brief Получение массива

\* \param message Мотивоционное сообщение для пользователя

\* \return Размер массива

\*/

size\_t GetSize(const string& message);

/\*\*

\* \brief Вывод элементов массива

\* \param size Размер массива

\* \return Строка со значениями индексов массива

\*/

string ToString(const int\*, const size\_t size);

/\*\*

\* \brief Функция для замены второго элемента массива на максимальный среди отрицательных

\* \param size Размер массива

\*/

void Replace(int\*, const size\_t size);

/\*\*

\* \brief Создание массива с добавленным числом К (по определённому правилу) вводимым пользователем

\* \param size Размер массива

\* \param new\_size Размер нового массива

\* \param k Число K вводимое пользователем

\* \return Возвращает массив, в котором перед элементами, значения которых имеют цифру 1 добавлено число К вводимое пользователем

\*/

int\* AddElement(int\* array, size\_t size, size\_t new\_size, const int k);

/\*\*

\* \brief Счетчик элементов, значения которых имеют цифру 1

\* \param size Размер массива

\* \return Возвращает количество элементов, значения которых имеют цифру 1

\*/

size\_t Сounter(int\* array, size\_t size);

/\*\*

\* \brief Формирует из массива D массив A по правилу (если номер четный, то значение элемента находится по формуле Ai = Di^2, а если нечетный, то по формуле Ai = Di / i)

\* \param size Размер массива

\* \return массив А

\*/

int\* FormArray(int\*, const size\_t size);

/\*\*

\* \brief Точка входа в программу

\* \return (Код ошибки 0) успех

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

auto error\_code = 0;

int\* my\_array = nullptr;

const int min\_value = -10;

const int max\_value = 10;

try

{

auto size = GetSize("Введите размер массива: ");

cout << "Выберите способ создания массива: " << static\_cast<int>(Choice::Manual) << " - вручную, " << static\_cast<int>(Choice::Random) << " - заполнить случайными числами ";

int selection;

cin >> selection;

my\_array = GetArray(size, selection, min\_value, max\_value);

cout << "Итоговый массив:\n";

cout << ToString(my\_array, size);

cout << "\nМассив после замены второго элемента массива на максимальный среди отрицательных:\n ";

Replace(my\_array, size);

cout << ToString(my\_array, size);

int k;

cout << "\nВведите число K: ";

cin >> k;

cout << "Массив после вставки числа К перед всеми элементами имеющими цифру 1:\n ";

size\_t counter\_add\_elements = Сounter(my\_array, size);

size\_t new\_size = size + counter\_add\_elements;

int\* new\_array = AddElement(my\_array, size, new\_size, k);

cout << ToString(new\_array, new\_size);

cout << "\nНовый массив:\n ";

int\* array\_a = FormArray(my\_array, size);

cout << ToString(array\_a, size);

if (new\_array != nullptr)

{

delete[] new\_array;

new\_array = nullptr;

}

if (array\_a != nullptr)

{

delete array\_a;

array\_a = nullptr;

}

}

catch (exception& e)

{

cout << e.what();

error\_code = 1;

}

if (my\_array != nullptr)

{

delete[] my\_array;

my\_array = nullptr;

}

return error\_code;

}

size\_t GetSize(const string& message)

{

int size = -1;

cout << message;

cin >> size;

if (size <= 0)

{

throw out\_of\_range("Неправильный размер массива");

}

return size;

}

int\* GetArray(const size\_t size, const int selection, const int min\_value, const int max\_value)

{

const auto array = new int[size];

//Will be used to obtain a seed for the random number engine

random\_device rd;

//Standard mersenne\_twister\_engine seeded with rd()

mt19937 gen(rd());

const uniform\_int\_distribution<> uniformIntDistribution(min\_value, max\_value);

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

{

switch (selection)

{

case static\_cast<int>(Choice::Manual):

{

cout << "Введите " << index + 1 << " элемент массива в диапозоне [-10;10]: ";

cin >> array[index];

break;

}

case static\_cast<int>(Choice::Random):

{

array[index] = uniformIntDistribution(gen);

break;

}

}

}

return array;

}

string ToString(const int\* array, const size\_t size)

{

if (array == nullptr)

throw invalid\_argument("Массив не существует");

stringstream buffer;

buffer << "{";

for (size\_t index = 0; index < size - 1; index++)

{

buffer << array[index] << ", ";

}

buffer << array[size - 1] << "}";

return buffer.str();

}

void Replace(int\* array, const size\_t size) {

int value = 0;

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

if (array[i] < value) {

value = array[i];

array[2] = value;

}

}

}

int\* AddElement(int\* array, size\_t size, size\_t new\_size, const int k) {

int\* new\_array = new int[new\_size];

size\_t index = 0;

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

int value = abs(array[i]);

if ((value % 10 == 1) || (value / 10 == 1) || (value / 10 == -1)) {

new\_array[i + index] = k;

new\_array[i + index + 1] = array[i];

index++;

}

else

{

new\_array[i + index] = array[i];

}

}

return new\_array;

}

size\_t Сounter(int\* array, size\_t size) {

size\_t counter\_add\_elements = 0;

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

int value = abs(array[i]);

if (value % 10 == 1 || value / 10 == 1 || value / 10 == -1) {

counter\_add\_elements++;

}

}

return counter\_add\_elements;

}

int\* FormArray(int\* array\_d, const size\_t size) {

int\* array\_a = new int[size];

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

if (i % 2 == 0) {

array\_a[i] = pow(array\_d[i], 2);

}

else {

array\_a[i] = array\_d[i] / i;

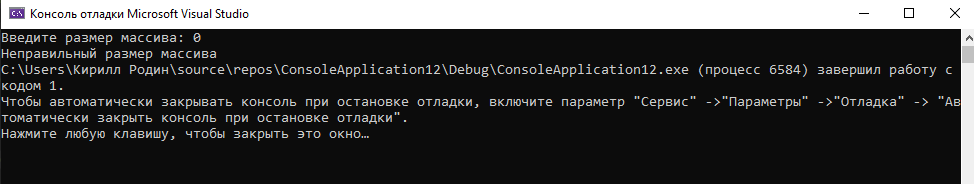
}

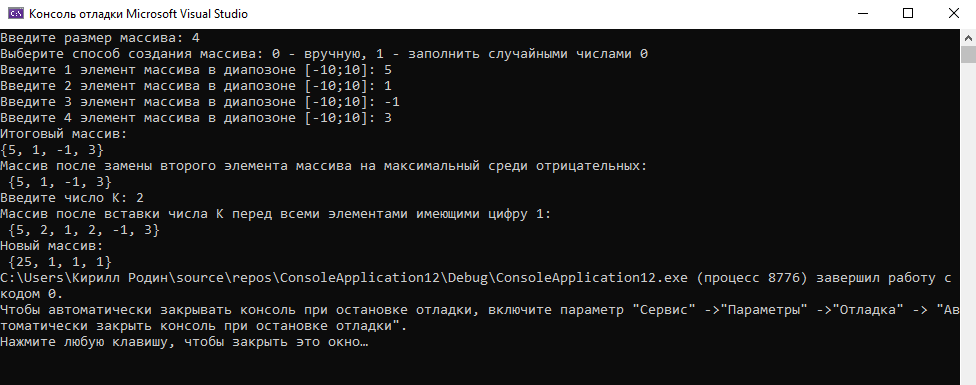
}

return array\_a;

}

## **Решение тестовых примеров на С++**





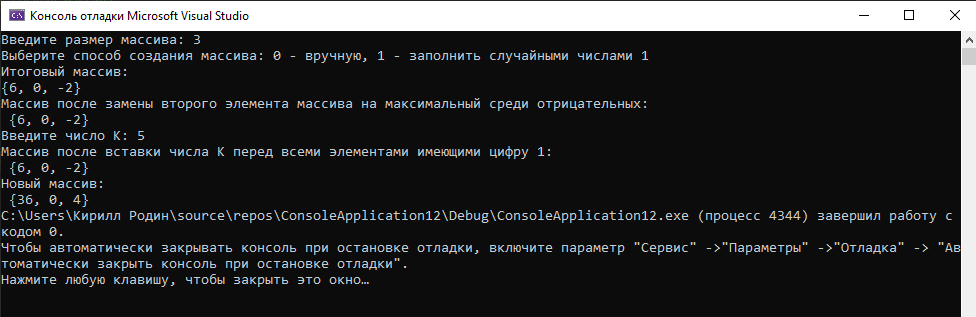


Рисунок 4 - Решение тестовых примеров для задания 4-2