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

Земсков Н.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Условие задачи 3](#_Toc72939989)

[Блок-схема алгоритма 4](#_Toc72939990)

[Код алгоритма на языке C++ 5](#_Toc72939991)

[Пример результата работы алгоритма 8](#_Toc72939992)

# 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.Найти сумму четных элементов.

2. Подсчитать количество элементов массива, значения которых состоят из двух цифр.  
3. Заменить последний отрицательный элемент массива на модуль первого элемента массива.

# Блок-схема алгоритма

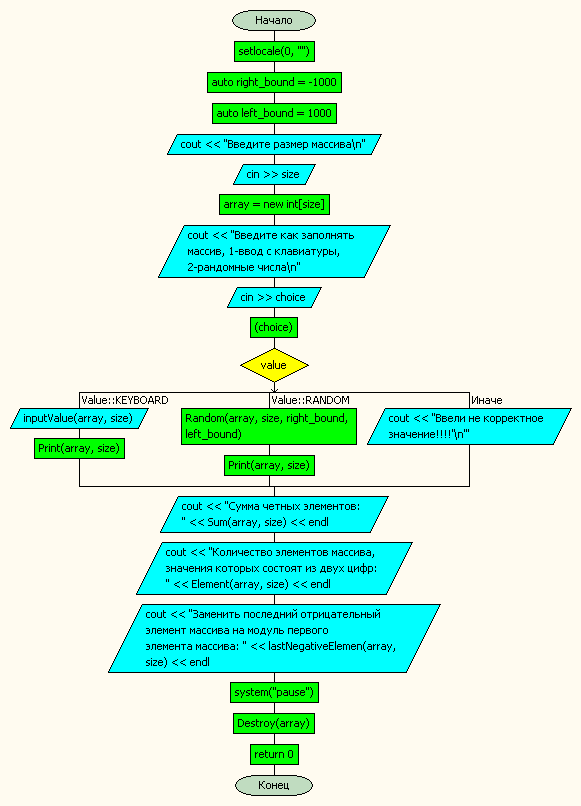
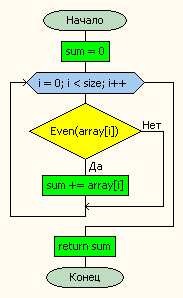
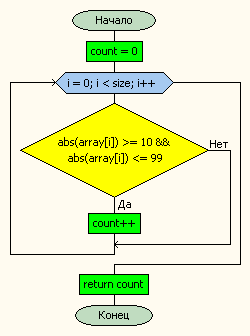


Рисунок 1 - Блок схема основной функции

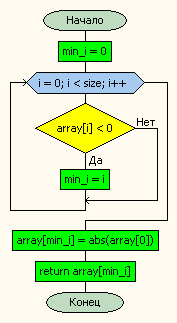


**Рисунок 2 - Блок-схема нахождения суммы четных элементов**



**Рисунок 3 - Блок-схема подсчитывающая количество элементов массива, значения которых состоят из двух цифр.**

# 



**Рисунок 4 - Блок-схема заменяющая последний отрицательный элемент массива на модуль первого элемента массива.**

# Код алгоритма на языке C++

#include <iostream>

#include <cmath>

#include <random>

#include <iomanip>

using namespace std;

/\*\*

\* \brief Заполнение массива с клавиатуры

\* \param array Массив

\* \param size Размер массива

\*/

void inputValue(int\* array, const int size);

/\*\*

\* \brief Заполнение массива случайными числами

\* \param array Массив

\* \param size Размер массива

\* \param right\_bound Минимальная граница

\* \param left\_bound Максимальная граница

\*/

void Random(int\* array, const int size, const int right\_bound, const int left\_bound);

/\*\*

\* \brief Вывод массива

\* \param array Массив

\* \param size Размер массива

\*/

void Print(int\* array, const int size);

/\*\*

\* \brief Проверка индекса элемента на четность

\* \param i Индекс массива

\*/

bool Even(const int i);

/\*\*

\* \brief Уничтожает массив

\* \param array Массив

\*/

void Destroy(int\* array);

/\*\*

\* \brief Находит сумму четных элементов масива

\* \param array Массив

\* \param size Размер масива

\* \\return Возвращает сумму чётных элементов масива

\*/

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

/\*\*

\* \brief Находит количество элементов массива, значения которых состоят из двух цифр.

\* \param array Массив

\* \param size Размер масива

\* \\return Возвращает количество элементов массива, значения которых состоят из двух цифр.

\*/

int Element(int\* array, const int size);

/\*\*

\* \brief Заменяет последний отрицательный элемент масива на модуль первого элемента массива.

\* \param array Массив

\* \param size Размер масива

\* \\return Возвращает последний отрицательный элемент массива на модуль первого элемента массива.

\*/

int lastNegativeElemen(int\* array, const int size);

enum class Value {

NONE,

KEYBOARD,

RANDOM,

};

int main() {

setlocale(0, "");

const auto right\_bound = -1000;

const auto left\_bound = 1000;

int size;

cout << "Введите размер массива\n";

cin >> size;

int\* array;

array = new int[size];

int choice;

cout << "Введите как заполнять массив, 1-ввод с клавиатуры, 2-рандомные числа\n";

cin >> choice;

const auto value = static\_cast<Value>(choice);

switch (value) {

case Value::KEYBOARD:

inputValue(array, size);

Print(array, size);

break;

case Value::RANDOM:

Random(array, size, right\_bound, left\_bound);

Print(array, size);

break;

default:

cout << "Ввели не корректное значение!!!!'\n'";

}

cout <<"Сумма четных элементов: " <<Sum(array, size) << endl;

cout <<"Количество элементов массива, значения которых состоят из двух цифр: " <<Element(array, size) << endl;

cout <<"Заменить последний отрицательный элемент массива на модуль первого элемента массива: " <<lastNegativeElemen(array, size) << endl;

system("pause");

Destroy(array);

return 0;

}

void inputValue(int\* array, const int size) {

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

cin >> array[i];

}

}

void Random(int\* array, const int size, const int right\_bound, const int left\_bound) {

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

std::random\_device rd;

std::mt19937 gen(rd());

const uniform\_int\_distribution<> uniformIntDistribution(right\_bound, left\_bound);

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

array[i] = uniformIntDistribution(gen);

}

}

}

void Print(int\* array, const int size) {

cout << "Массив [" << size << "]" << "= {";

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

cout << array[i] << "; " << setw(3);

}

cout << array[size - 1] << "}" << endl;

}

void Destroy(int\* array) {

if (array != nullptr) {

delete[] array;

array = nullptr;

}

}

bool Even(const int a) {

return a % 2 == 0;

}

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

double sum = 0;

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

if (Even(array[i]))

sum += array[i];

}

return sum;

}

int Element(int\* array, const int size) {

int count = 0;

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

if (abs(array[i]) >= 10 && abs(array[i]) <= 99)

count++;

}

return count;

}

int lastNegativeElemen(int\* array, const int size) {

int min\_i = 0;

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

if (array[i] < 0)

min\_i = i;

}

array[min\_i] = abs(array[0]);

return array[min\_i];

}

# Пример результата работы алгоритма

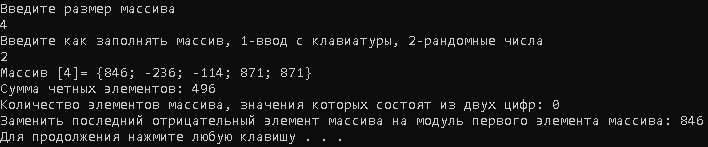


Рисунок 2 - Пример работы программы со случайными числами

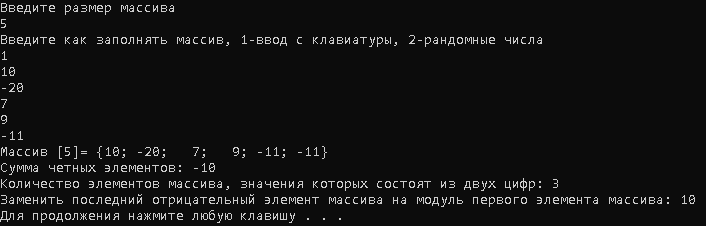


Рисунок 3 - Пример работы программы с введёнными числами