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 – 211

Земсков Н.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[Условие задачи 3](#_Toc72945088)

[Блок-схема алгоритма 4](#_Toc72945089)

[Код алгоритма на языке C++ 13](#_Toc72945090)

[Пример результата работы алгоритма 16](#_Toc72945091)

[Список использованной литературы 17](#_Toc72945092)

# 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

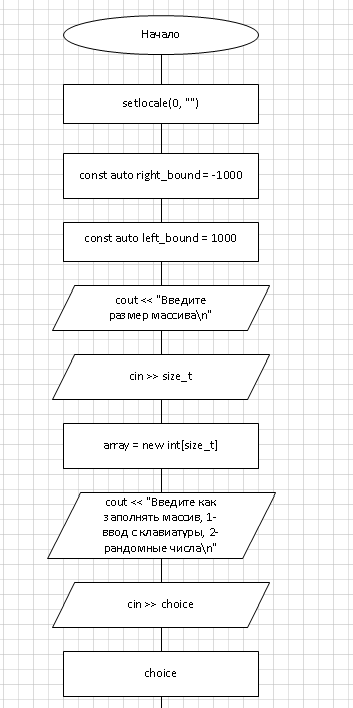


Рисунок 1 - Блок-схема ф-ции main

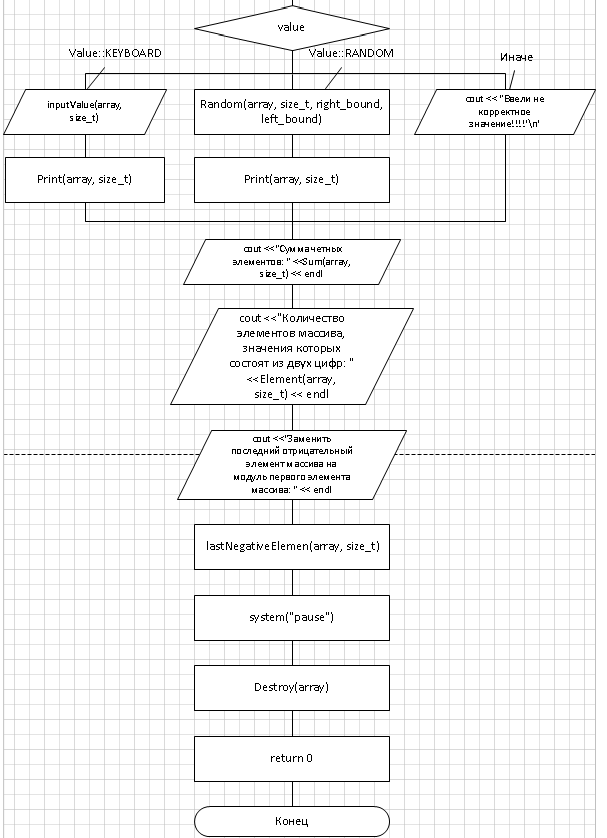


Рисунок 2 - Блок-схема ф-ции main (продолжение)

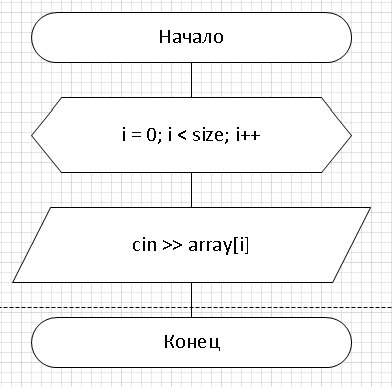


Рисунок 3 – ф-ия ввода значения

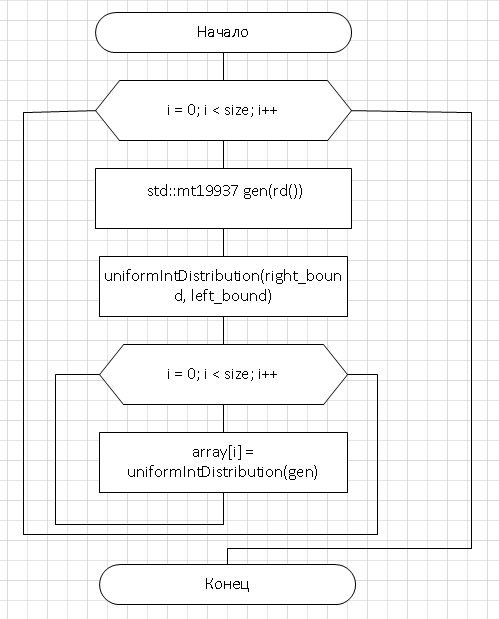


Рисунок 4 - Функция Рандома

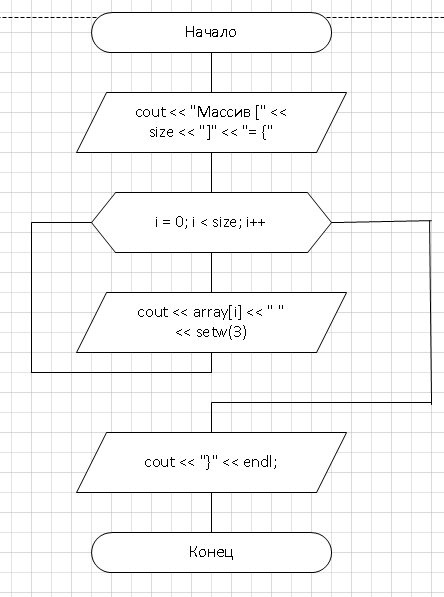
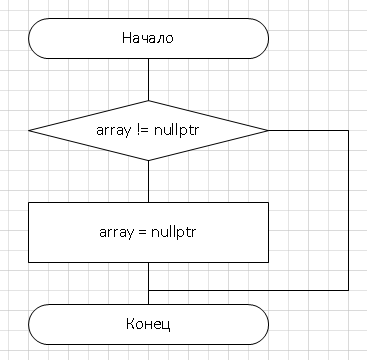


Рисунок 5 – Печатает массив



Да

Нет

Рисунок 6 – Уничтожает массив

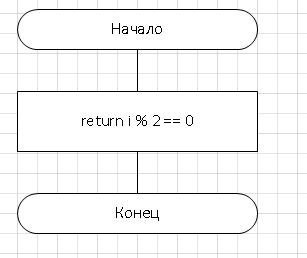
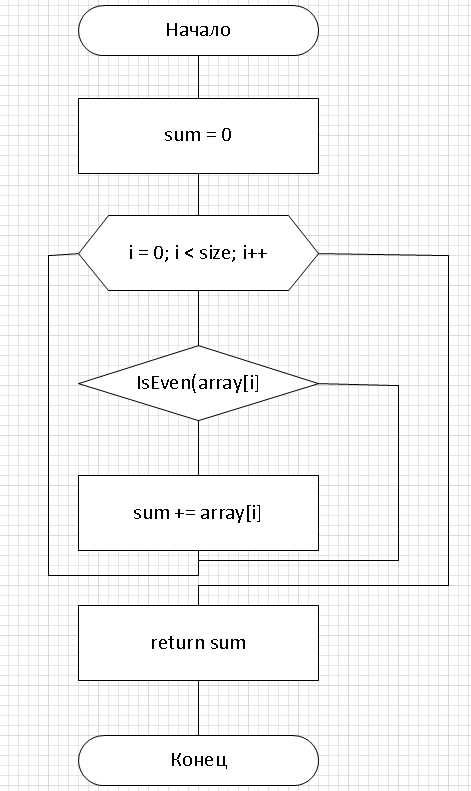


Рисунок 7 – Вычисления



Да

Нет

Рисунок 8 – Вычисляет сумму чётных элементов

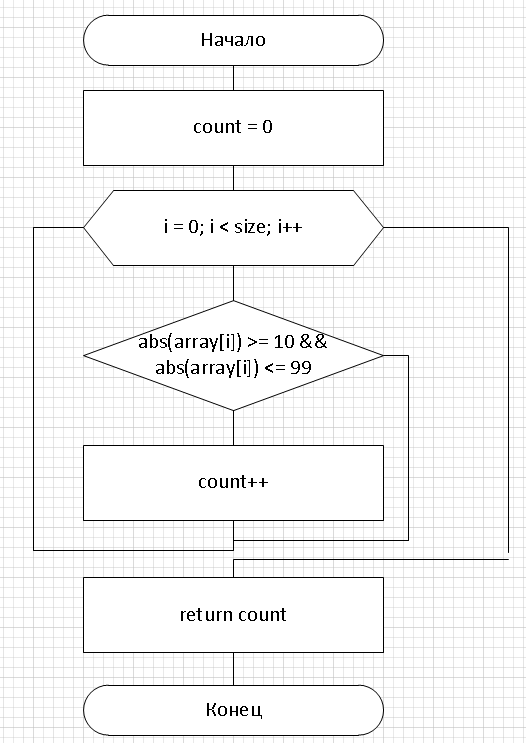
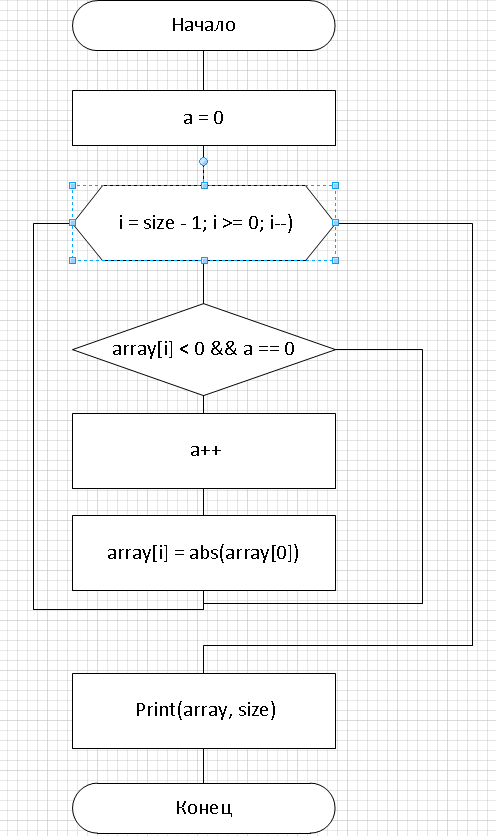


Рисунок 9 – Вычисление суммы элементов из двух цифр



Нет

Рисунок 10 – Замена последнего отрицательного элемента, первым по модулю

# 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 IsEven(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.

\*/

void 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\_t;

cout << "Введите размер массива\n";

cin >> size\_t;

int\* array;

array = new int[size\_t];

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\_t);

Print(array, size\_t);

break;

case Value::RANDOM:

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

Print(array, size\_t);

break;

default:

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

}

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

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

cout <<"Заменить последний отрицательный элемент массива на модуль первого элемента массива: " << endl;

lastNegativeElemen(array, size\_t);

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 << "}" << endl;

}

void Destroy(int\* array) {

if (array != nullptr) {

delete[] array;

array = nullptr;

}

}

bool IsEven(const int i) {

return i % 2 == 0;

}

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

double sum = 0;

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

if (IsEven(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;

}

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

int a = 0;

for (int i = size - 1; i >= 0; i--) {

if (array[i] < 0 && a == 0){

a++;

array[i] = abs(array[0]);

}

}

Print(array, size);

}

# Пример результата работы алгоритма

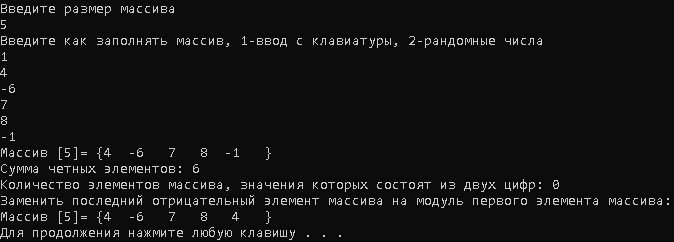


Рисунок 11 - пример работы программы (заполнение с клавиатуры)

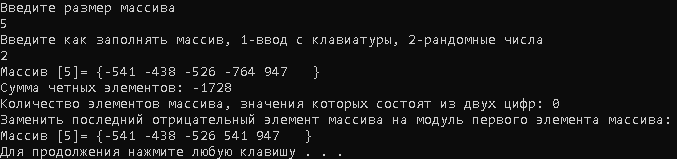


Рисунок 12 - пример работы программы (заполнение случайными числами)

# Список использованной литературы

1. URL: http://coldfox.ru/article/5ab95d2350e0790c10afcc00/Подробное-руководство-по-Git
2. URL: https://ru.wikipedia.org/wiki/Моменты\_случайной\_величины
3. URL: https://code-live.ru/post/cpp-arrays/