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 – 112

Потапов А.К.

Проверил: к.т.н., доц.

Васильева М.А.

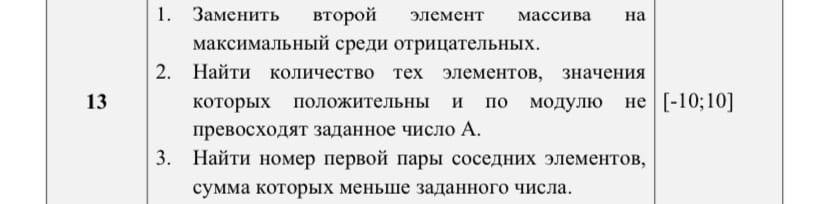
Москва 2021

**Содержание**

Задание 4-1……………………………………. 2

**Задание 4-1**

1. **Формулировка задачи**
2. Найти сумму элементов, значения которых состоят из одной цифры.
3. Заменить элементы массива между минимальным и максимальным на те же элементы в обратном порядке
4. Найти номер последней пары соседних элементов с одинаковыми знаками, произведение которых меньше заданного числа.



1. **Блок-схема алгоритма**

Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 1, Рисунок 2, Рисунок 3).

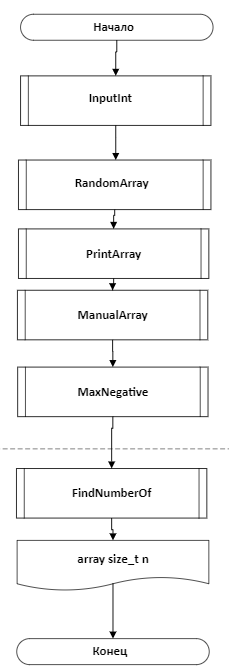


Рисунок 1 – Блок-схема функции main()

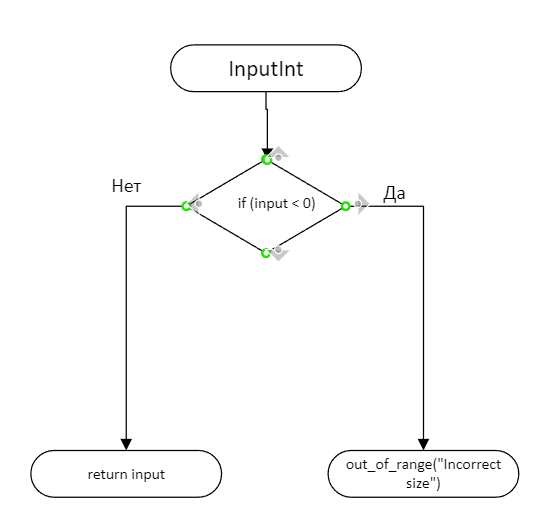


Рисунок 2 – Блок-схема функции InputInt

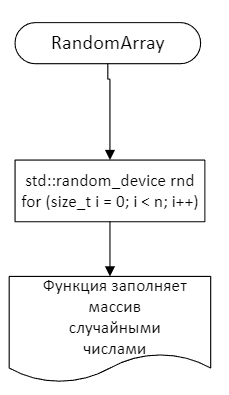


Рисунок 3 – Блок-схема функции RandomArray

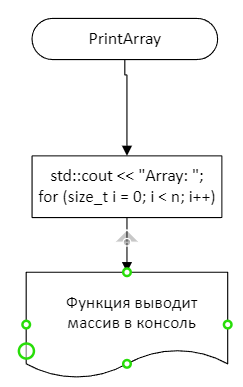


Рисунок 4 – Блок-схема функции PrintArray

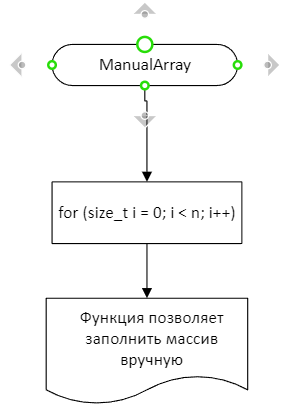


Рисунок 5 – Блок-схема функции ManualArray

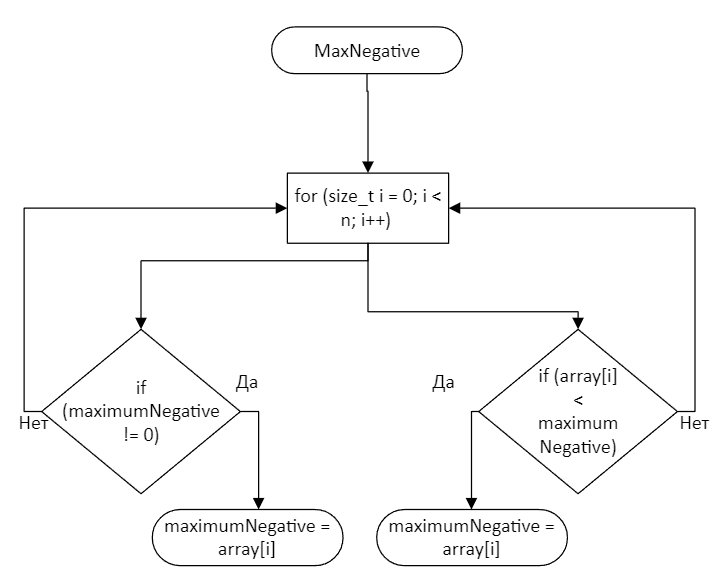


Рисунок 6 – Блок-схема функции MaxNegative

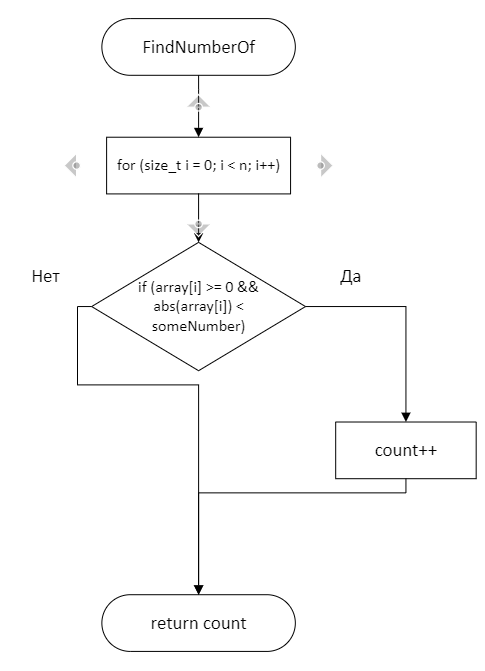


Рисунок 7 – Блок-схема функции FindNumberOf

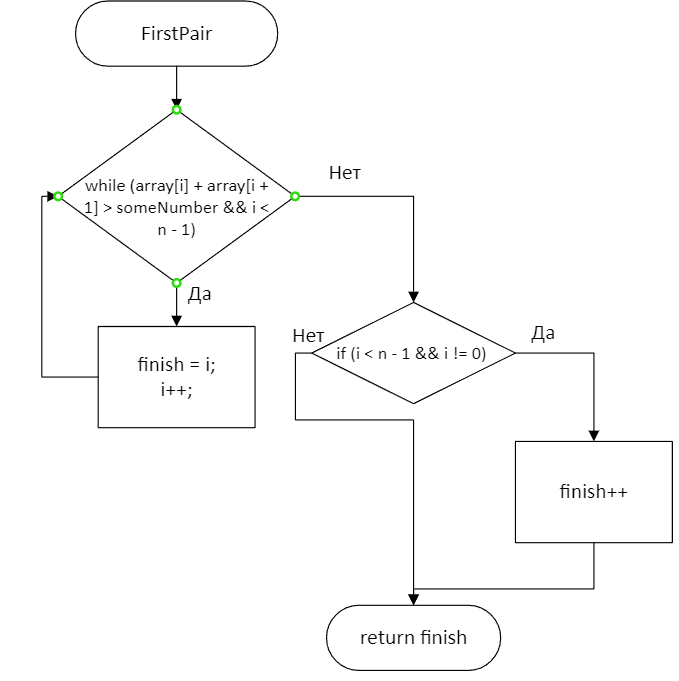


Рисунок 8 – Блок-схема функции FirstPair

1. **Решение задачи на языке программирования C++**

Код программы

#include <iostream>

#include <random>

/\*\*

\*\brief Функция возвращает целочисленное число, введённое пользователем

\*\return Возвращает число, введённое пользователем

\*/

int InputInt(const std::string& message);

/\*\*

\*\brief Функция заполняет массив случайными числами

\*\param array массив

\*\param n размер Массива

\*/

void RandomArray(int\* array, const size\_t n);

/\*\*

\*\brief Функция выводит массив в консоль

\*\param array массив

\*\param n размер Массива

\*/

void PrintArray(int\* array, const size\_t n);

/\*\*

\*\brief Функция позволяет заполнить массив вручную

\*\param array массив

\*\param n размер Массива

\*/

void ManualArray(int\* array, const size\_t n);

/\*\*

\*\brief Функция меняет второй элемент массива на максимальный отрицательный в массиве

\*\param array массив

\*\param n размер Массива

\*/

void MaxNegative(int\* array, const size\_t n);

/\*\*

\*\brief Функция считает количество элементов в массиве, положительных и по модулю не превосходящих заданное число

\*\param array массив

\*\param n размер Массива

\*\return Возвращает количество элементов

\*/

int FindNumberOf(int\* array, const size\_t n);

/\*\*

\*\brief Функция ищет индексы первой пары чисел в массиве, сумма которых меньше заданного числа

\*\param array массив

\*\param n размер Массива

\*\return Возвращает индекс первого элемента пары в случае успеха

\*/

int FirstPair(int\* array, const size\_t n);

/\*\*

\*\brief Класс констант метода ввода

\*/

enum class Input

{

random,

manual

};

/\*\*

\*\brief Точка входа в программу

\*\return 0 в случае успеха

\*/

int main()

{

size\_t n = InputInt("Enter number of elements in array");

std::cout << static\_cast<int>(Input::random) << " — random\n"

<< static\_cast<int>(Input::manual) << " — manual";

int\* array = new int[n];

size\_t choose = InputInt("");

try {

auto chose = static\_cast<Input>(choose);

switch (chose)

{

case Input::random:

{

RandomArray(array, n);

PrintArray(array, n);

break;

}

case Input::manual:

{

ManualArray(array, n);

PrintArray(array, n);

break;

}

}

}

catch (std::out\_of\_range&) {

return 1;

}

std::cout << "Number of elements, which positive and abs < some number\n" << FindNumberOf(array, n) << " — nuber of elements\n";

std::cout << "Numbers of first pair, which sum < some number\n";

int temp = FirstPair(array, n);

std::cout << "Indexes of first pair — " << temp << ", " << temp + 1 << std::endl;

std::cout << "Replace 2nd element in array by maximum negative\n";

MaxNegative(array, n);

PrintArray(array, n);

return 0;

}

void PrintArray(int\* array, const size\_t n)

{

std::cout << "Array: ";

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

{

std::cout << array[i] << " ";

}

std::cout << std::endl;

}

void RandomArray(int\* array, const size\_t n)

{

std::random\_device rnd;

std::mt19937 gen(rnd());

std::uniform\_int\_distribution<> Uniform\_int\_distribution(-10, 10);

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

{

array[i] = Uniform\_int\_distribution(gen);

}

}

void ManualArray(int\* array, const size\_t n)

{

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

{

std::cin >> array[i];

}

}

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

int maximumNegative = 0;

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

if (array[i] < maximumNegative) {

maximumNegative = array[i];

}

}

if (maximumNegative != 0) {

array[1] = maximumNegative;

}

}

int FindNumberOf(int\* array, const size\_t n) {

std::cout << "Enter number A: ";

int someNumber;

std::cin >> someNumber;

size\_t count = 0;

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

if (array[i] >= 0 && abs(array[i]) < someNumber) {

count++;

}

}

return count;

}

int FirstPair(int\* array, const size\_t n) {

std::cout << "Enter number A: ";

int someNumber;

std::cin >> someNumber;

int finish = 0; //индекс первого элемента пары

size\_t i = 0;

while (array[i] + array[i + 1] > someNumber && i < n - 1) {

finish = i;

i++;

}

if (i < n - 1 && i != 0) {

finish++;

}

return finish;

}

int InputInt(const std::string& message)

{

std::cout << message << std::endl;

int input = -1;

std::cin >> input;

if (input < 0) {

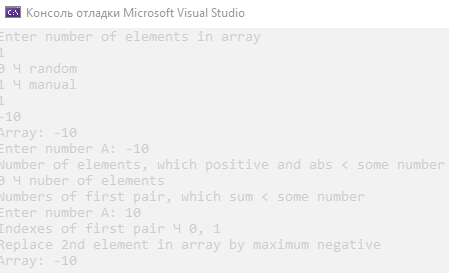
throw (std::out\_of\_range("Incorrect size"));

}

return input;

}

1. **Решение тестовых примеров**



1. **Зачет задания в GitHub**

