МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

Программирование

Лабораторная работа № 5

Тема

Изучение приемов работы с одномерными массивами в языке C++.

Выполнил

Студент гр. Б260221

Воробьёв А.А.

Проверила

Сафронова М.А.

Тула 2023 г.

**Цель работы:** Задача лабораторной заключается в изучении приемов работы с одномерными массивами в языке C++. Одномерные массивы являются одной из основных структур данных в программировании, и их эффективное использование позволяет решать различные задачи, связанные с хранением и обработкой множества элементов одного типа.

При изучении приемов работы с одномерными массивами в языке C++ целью работы может быть овладение следующими навыками:

1. Объявление и инициализация массивов: изучение синтаксиса объявления одномерных массивов в C++ и их инициализации значениями. Это включает понимание использования квадратных скобок, указания размерности массива и присваивания значений элементам.
2. Доступ к элементам массива: изучение способов доступа к элементам массива по индексу. Понимание того, что индексация массива начинается с нуля, и использование циклов или явного указания индекса для доступа к элементам.
3. Обход массива: изучение различных методов обхода одномерных массивов, таких как использование циклов for или while для последовательного доступа ко всем элементам массива.
4. Манипуляции с массивами: изучение методов изменения значений элементов массива, таких как присваивание новых значений по индексу или обмен значений между элементами массива.
5. Использование функций с массивами: изучение передачи массивов в функции и возврата массивов из функций. Понимание передачи массивов по значению и по ссылке, а также обработки массивов внутри функций.
6. Работа с динамическими массивами: изучение создания и управления динамическими массивами в C++, используя операторы new и delete. Понимание особенностей работы с динамическими массивами, включая выделение и освобождение памяти.
7. Работа с библиотеками и алгоритмами: изучение использования стандартных библиотек и алгоритмов C++ для работы с одномерными массивами. Это включает изучение функций, таких как std::sort для сортировки массива, std::find для поиска элемента в массиве и других полезных функций.

Цель работы состоит в том, чтобы овладеть всеми вышеперечисленными навыками и научиться эффективно работать с одномерными массивами в языке C++. Это позволит вам создавать более сложные программы, которые требуют обработки и управления большим количеством данных.

**Задание 1. Основные операции при работе с массивами**

Разработать программу на языке C++ для решения следующей задачи.

Задан массив вещественных чисел X(n). Найти

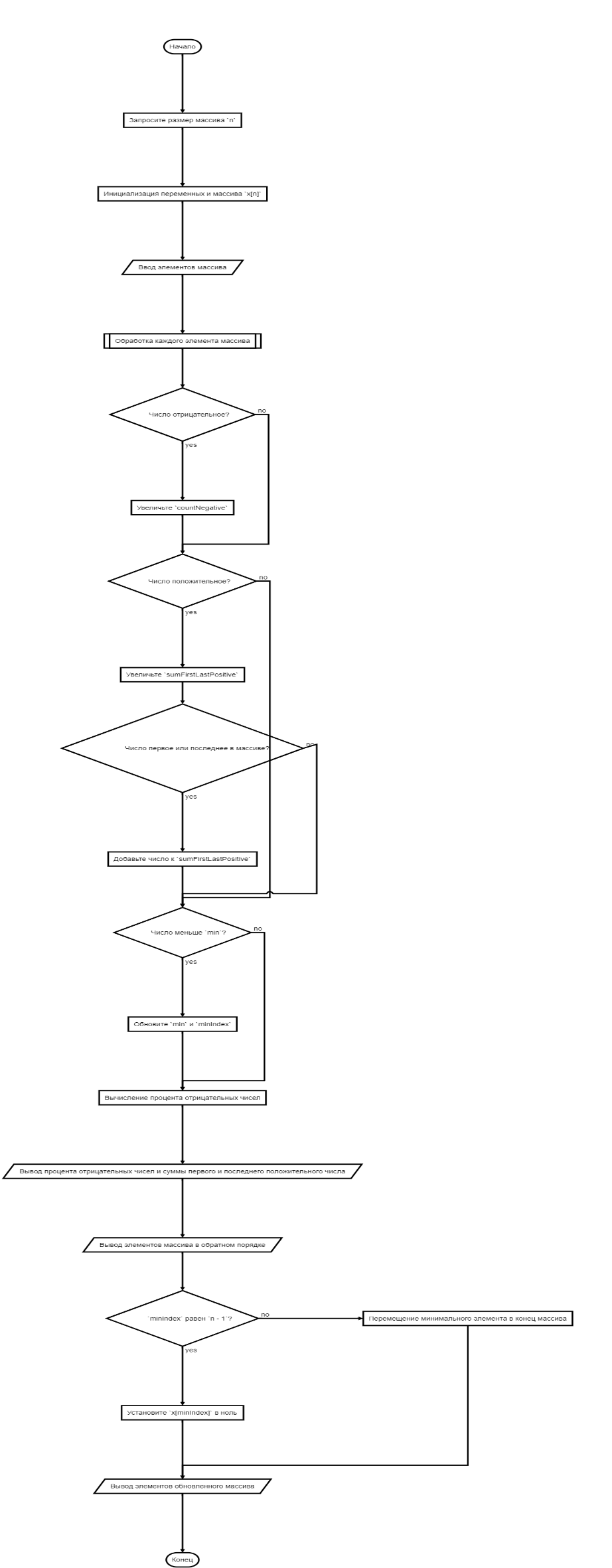
• процент отрицательных чисел в массиве;

• сумму первого и последнего положительных элементов.

Записать элементы заданного массива в обратном порядке. Определить положение

минимального элемента до и после преобразования. Удалить минимальный элемент.

**Блок-схема:**

****

**Код программы:**

void Task1()

{

int n;

cout << "Введите размер массива: ";

cin >> n;

double x[n];

int countNegative = 0;

double sumFirstLastPositive = 0;

double min = x[0];

int minIndex = 0;

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < n; i++)

{

cin >> x[i];

if (x[i] < 0)

{

countNegative++;

}

else if (x[i] > 0)

{

sumFirstLastPositive += x[i];

if (i == 0 || i == n - 1)

{

sumFirstLastPositive += x[i];

}

}

if (x[i] < min)

{

min = x[i];

minIndex = i;

}

}

double percentNegative = countNegative \* 100.0 / n;

cout << "Процент отрицательных чисел в массиве: " << percentNegative << "%" << endl;

cout << "Сумма первого и последнего положительных элементов: " << sumFirstLastPositive << endl;

for (int i = n - 1; i >= 0; i--)

{

cout << x[i] << " ";

}

cout << endl;

if (minIndex == n - 1)

{

// Минимальный элемент уже был последним

x[minIndex] = 0;

}

else

{

// Переставляем минимальный элемент в конец массива

for (int i = minIndex; i < n - 1; i++)

{

x[i] = x[i + 1];

}

x[n - 1] = min;

}

cout << "Массив после удаления минимального элемента:";

for (int i = 0; i < n - 1; i++)

{

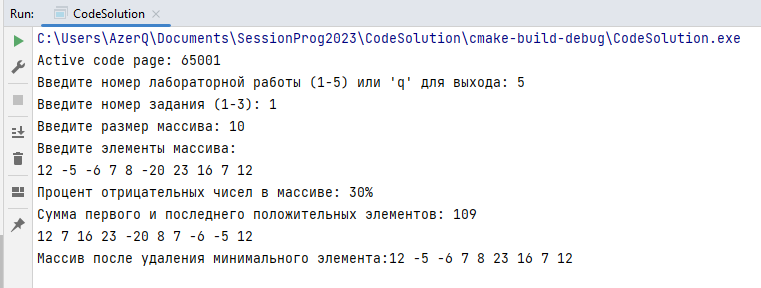
cout << x[i] << " ";

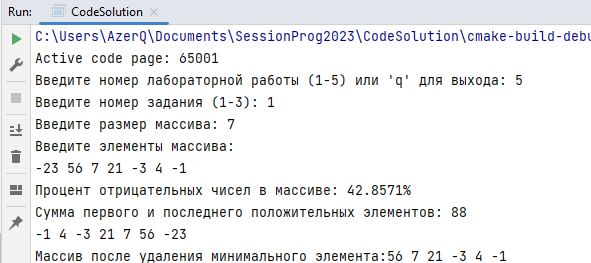
}

cout << endl;

}

**Вывод программы:**





**Задание 2. Применение функций для обработки массивов**

Разработать программу на языке C++ для решения следующей задачи.

Задан массив целых чисел X(n). Шесть наибольших чисел этого массива переписать в массив Z. Удалить из массива Z все чётные числа. Вывести на экран элементы массива Z в восьмеричной системе счисления.

**Блок-схема:**

****

**Код программы:**

void Task2()

{

int n = 0;

cout << "Размер массива: ";

cin >> n;

int arr[n];

cout << "Введите массив: ";

for (int i = 0; i < n; i++)

{

cin >> arr[i];

}

MaxNumbers(arr, n);

}

void MaxNumbers(int mass[], int size)

{

/\* Отсортируем массив по убыванию \*/

for (int i = 1; i < size; ++i)

{

for (int j = 0; j < size - i; j++)

{

if (mass[j] < mass[j + 1])

{

// Обмен местами

int temp = mass[j];

mass[j] = mass[j + 1];

mass[j + 1] = temp;

}

}

}

int Z[6];

int oddCount = 0;

for (int i = 0; i < 6; i++)

{

Z[i] = mass[i];

if (Z[i] % 2 == 0)

{

oddCount++;

}

}

cout << "Массив 6 максимальных элементов: " << endl;

PrintArray(Z, 6, false);

int oddZ[oddCount];

int oddIterator = 0;

for (int i = 0; i < 6; i++)

{

if (Z[i] % 2 == 0)

{

oddZ[oddIterator++] = Z[i];

}

}

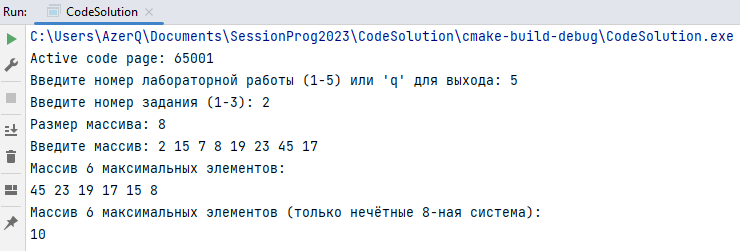
cout << endl

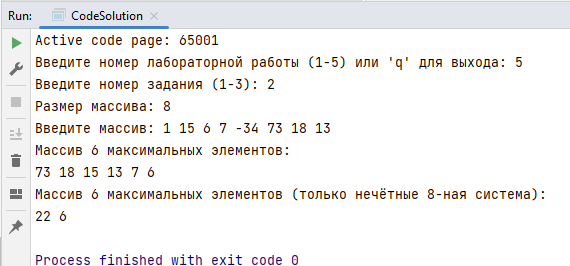
<< "Массив 6 максимальных элементов (только нечётные 8-ная система): " << endl;

PrintArray(oddZ, oddCount, true);

}

**Вывод программы:**

****

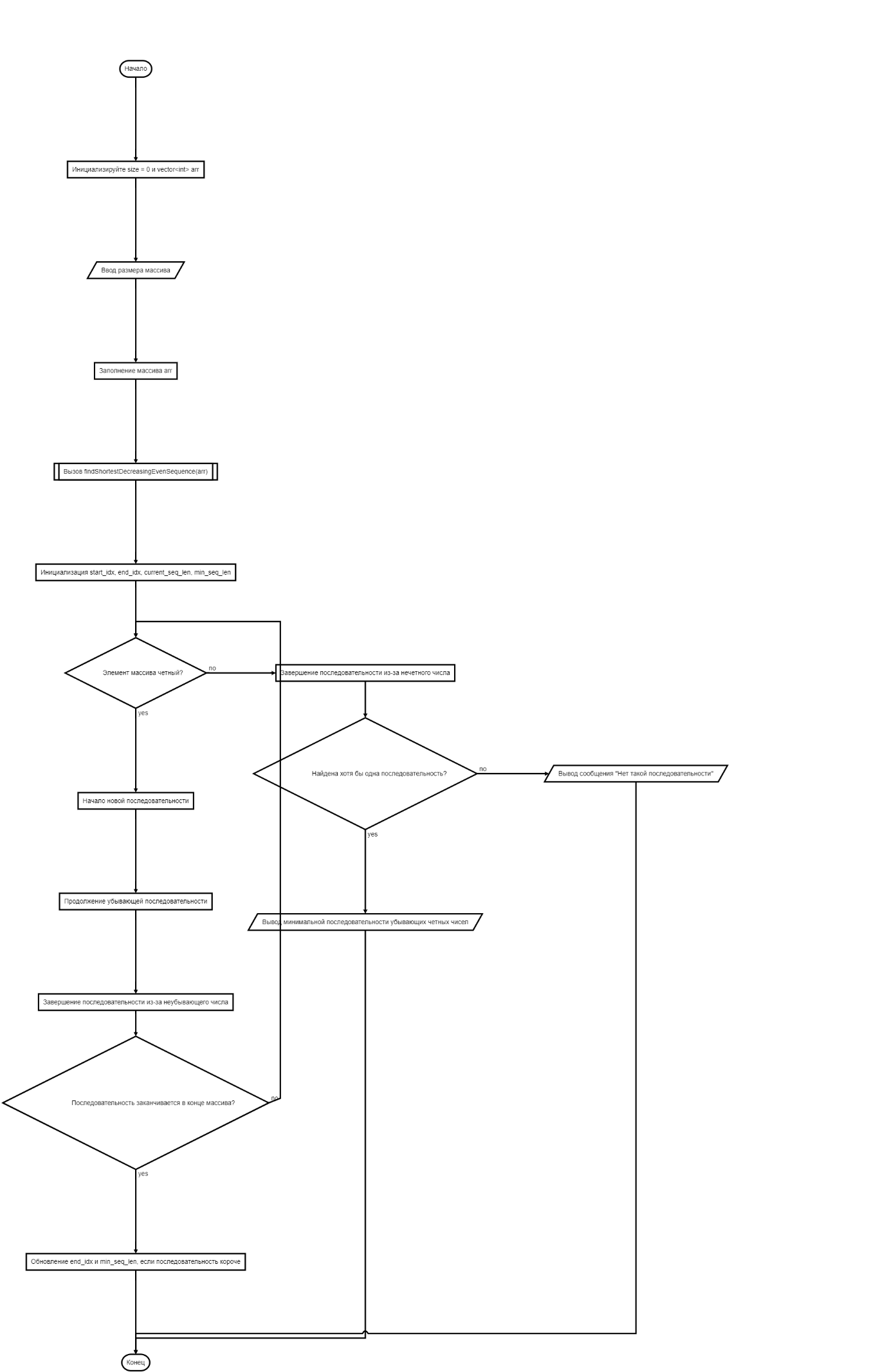
****

**Задание 3. Работа с группами элементов в массиве**

Разработать программу на языке C++ для решения следующей задачи.

В массиве целых чисел найти группу наименьшей длины, которая состоит из убывающей последовательности чётных чисел.

**Блок-схема:**

****

**Код программы:**

void Task3()

{

int size = 0;

std::vector<int> arr;

cout << endl

<< "Введите размер массива: ";

cin >> size;

for (int i = 0; i < size; i++)

{

int elem;

cin >> elem;

arr.push\_back(elem);

}

findShortestDecreasingEvenSequence(arr);

}

void findShortestDecreasingEvenSequence(std::vector<int> arr)

{

int start\_idx = -1, end\_idx = -1, current\_seq\_len = 0, min\_seq\_len = arr.size() + 1;

for (int i = 0; i < arr.size(); i++)

{

if (arr[i] % 2 == 0)

{ // если встретилось четное число

if (current\_seq\_len == 0)

{ // начинаем новую последовательность

start\_idx = i;

current\_seq\_len = 1;

}

else if (arr[i] < arr[i - 1])

{ // продолжаем убывающую последовательность

current\_seq\_len++;

}

else

{ // обнаружили неубывающее число, заканчиваем последовательность

current\_seq\_len = 0;

}

if (current\_seq\_len > 0 && i == arr.size() - 1)

{ // последовательность заканчивается в конце массива

end\_idx = i;

if (current\_seq\_len < min\_seq\_len)

{

min\_seq\_len = current\_seq\_len;

}

break;

}

}

else

{ // обнаружили нечетное число, заканчиваем последовательность

if (current\_seq\_len > 0)

{

end\_idx = i - 1;

if (current\_seq\_len < min\_seq\_len)

{

min\_seq\_len = current\_seq\_len;

}

current\_seq\_len = 0;

}

}

}

// выводим результаты

if (min\_seq\_len == arr.size() + 1)

{ // не нашли ни одной последовательности

std::cout << "Нет такой последователльности" << std::endl;

}

else

{

std::cout << "Минимальная последователность убывающих чётных чисел: " << std::endl;

for (int i = start\_idx; i <= end\_idx; i++)

{

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

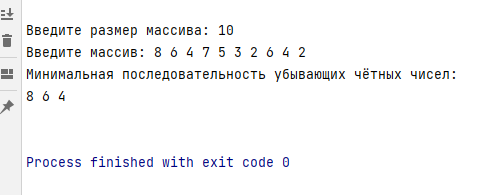
}

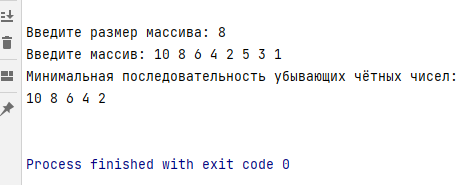
std::cout << std::endl;

}

}

**Вывод программы:**





**Вывод**

В данной работе были рассмотрены задачи по обработке одномерных массивов.

1. В первой задаче мы считали процент отрицательных элементов в массиве, суммировали первый и последний положительные элементы массива и удаляли минимальный элемент из массива, перемещая его в конец массива перед удалением.
2. Во второй задаче мы отсортировали массив по убыванию и выбрали 6 наибольших элементов, создав из них новый массив. Далее, из этих 6 элементов мы выбрали только четные числа и создали из них еще один массив.
3. В третьей задаче мы нашли в массиве группу наименьшей длины, которая состоит из убывающей последовательности четных чисел.

Все эти задачи были решены на языке программирования C++, были приведены блок-схемы алгоритмов и предоставлены примеры входных данных для тестирования программ.

Выводы: работа с массивами включает в себя множество различных операций, включая сортировку, поиск, выборку и другие. Решение этих задач требует знания и понимания как основ программирования, так и некоторых дополнительных концепций, включая условные операторы и циклы. Умение корректно работать с массивами - важный навык для любого программиста.