

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
Воробьев 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.

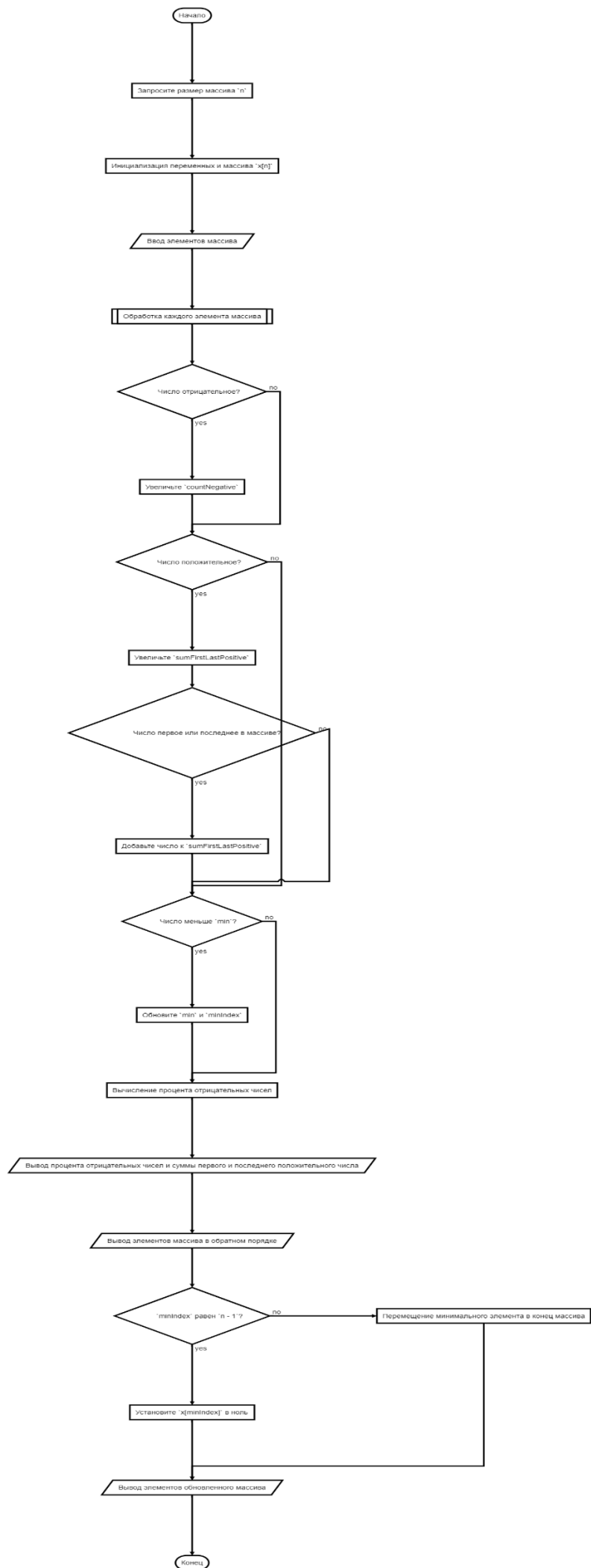
Задан массив вещественных чисел $X(n)$. Найти

- 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] << " ";
}

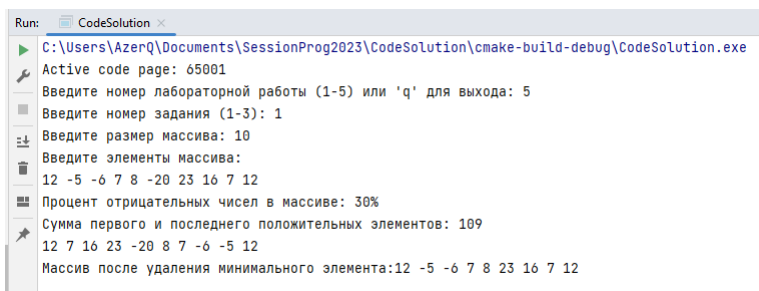
cout << endl;

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:



Run: CodeSolution

C:\Users\AzerQ\Documents\SessionProg2023\CodeSolution\cmake-build-debug\CodeSolution.exe

Active code page: 65001

Введите номер лабораторной работы (1-5) или 'q' для выхода: 5

Введите номер задания (1-3): 1

Введите размер массива: 10

Введите элементы массива:

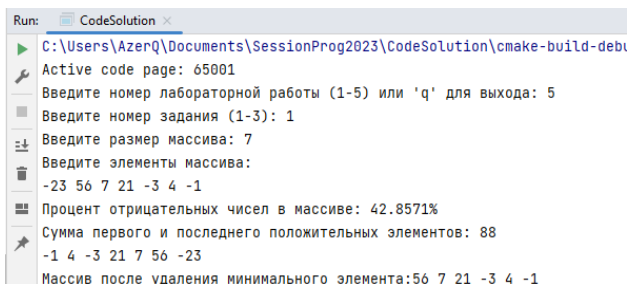
12 -5 -6 7 8 -20 23 16 7 12

Процент отрицательных чисел в массиве: 30%

Сумма первого и последнего положительных элементов: 109

12 7 16 23 -20 8 7 -6 -5 12

Массив после удаления минимального элемента: 12 -5 -6 7 8 23 16 7 12



Run: CodeSolution

C:\Users\AzerQ\Documents\SessionProg2023\CodeSolution\cmake-build-deb

Active code page: 65001

Введите номер лабораторной работы (1-5) или 'q' для выхода: 5

Введите номер задания (1-3): 1

Введите размер массива: 7

Введите элементы массива:

-23 56 7 21 -3 4 -1

Процент отрицательных чисел в массиве: 42.8571%

Сумма первого и последнего положительных элементов: 88

-1 4 -3 21 7 56 -23

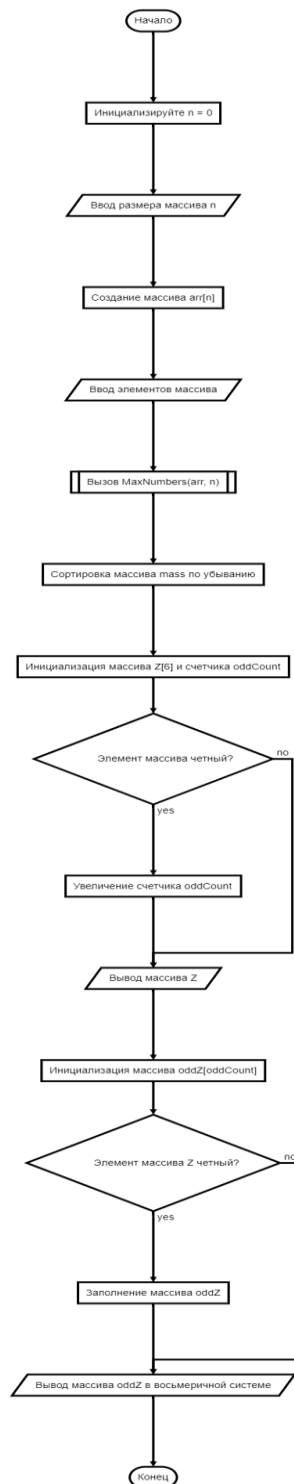
Массив после удаления минимального элемента: 56 7 21 -3 4 -1

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];
```



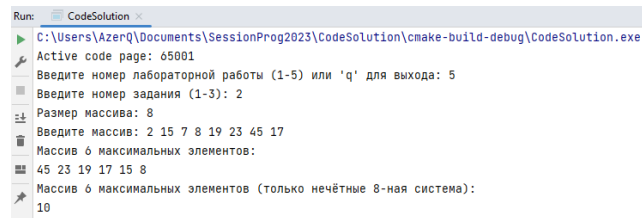
```

        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);
    }
}

```

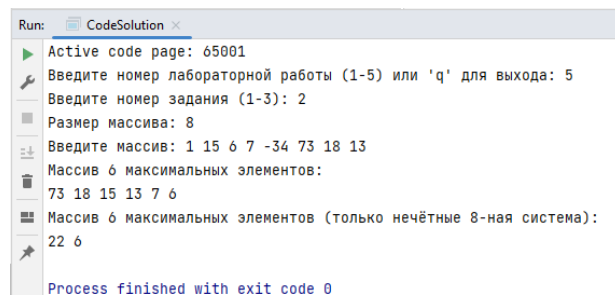
Вывод программы:



```

Run: CodeSolution
C:\Users\AzerQ\Documents\SessionProg2023\CodeSolution\cmake-build-debug\CodeSolution.exe
Active code page: 65001
Введите номер лабораторной работы (1-5) или 'q' для выхода: 5
Введите номер задания (1-3): 2
Размер массива: 8
Введите массив: 2 15 7 8 19 23 45 17
Массив 6 максимальных элементов:
45 23 19 17 15 8
Массив 6 максимальных элементов (только нечётные 8-ная система):
10

```



```

Run: CodeSolution
Active code page: 65001
Введите номер лабораторной работы (1-5) или 'q' для выхода: 5
Введите номер задания (1-3): 2
Размер массива: 8
Введите массив: 1 15 6 7 -34 73 18 13
Массив 6 максимальных элементов:
73 18 15 13 7 6
Массив 6 максимальных элементов (только нечётные 8-ная система):
22 6
Process finished with exit code 0

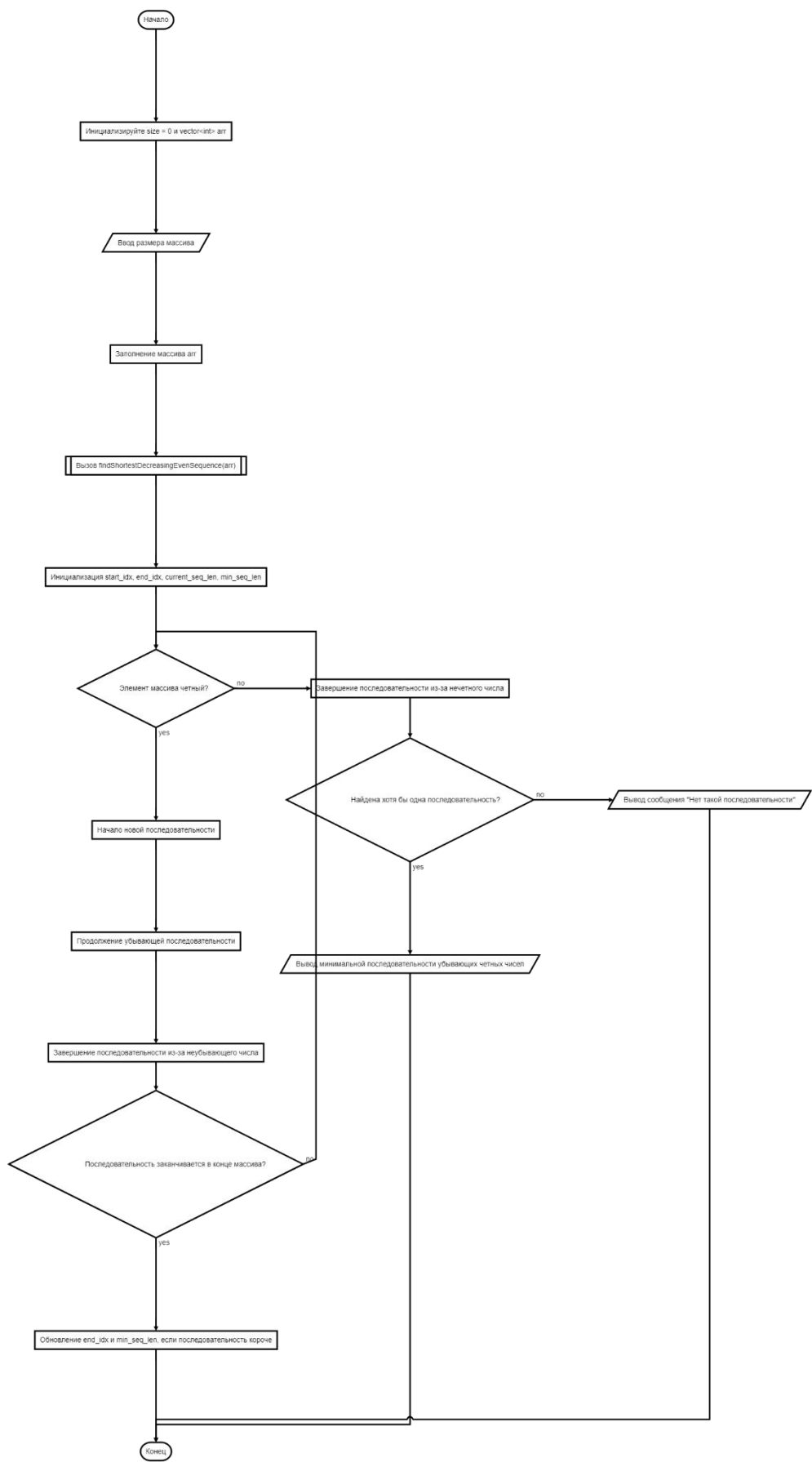
```

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;
    }
}
```

```

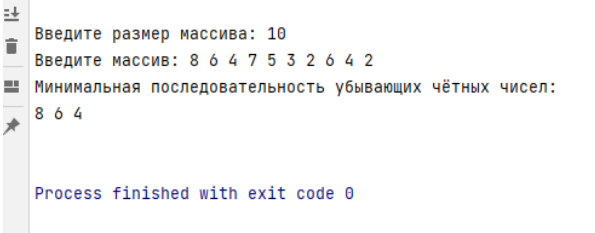
    }
}
else
{ // обнаружили нечетное число, заканчиваем последовательность
    if (current_seq_len > 0)
    {
        end_idx = i - 1;
        if (current_seq_len < min_seq_len)
        {
            min_seq_len = current_seq_len;
        }
        current_seq_len = 0;
    }
}
}

// выводим результаты
if (min_seq_len == arr.size() + 1)
{ // не нашли ни одной последовательности
    std::cout << "Нет такой последовательности" << std::endl;
}
else
{
    std::cout << "Минимальная последовательность убывающих чётных чисел: " <<
std::endl;

    for (int i = start_idx; i <= end_idx; i++)
    {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}
}
}

```

Вывод программы:

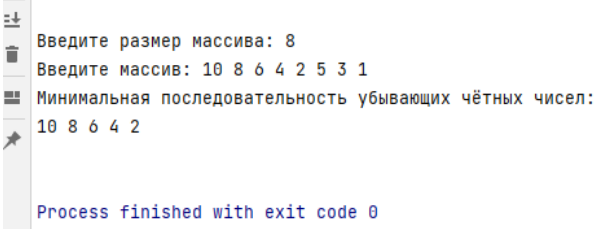


```

Введите размер массива: 10
Введите массив: 8 6 4 7 5 3 2 6 4 2
Минимальная последовательность убывающих чётных чисел:
8 6 4

Process finished with exit code 0

```

```

Введите размер массива: 8
Введите массив: 10 8 6 4 2 5 3 1
Минимальная последовательность убывающих чётных чисел:
10 8 6 4 2

Process finished with exit code 0

```

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.