Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 10

По дисциплине «Основы алгоритмизации и программирования»

На тему «Одномерные массивы»

Выполнил:

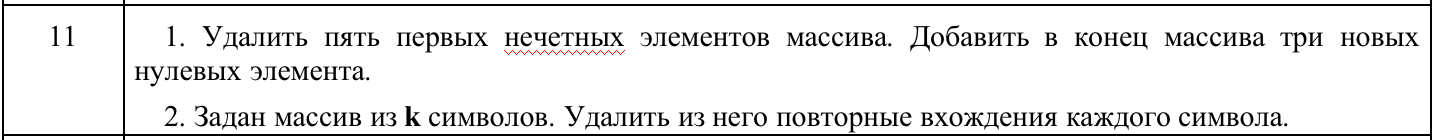
Студент 1 курса 9 группы

Павлович Ян Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

1. В соответствии со своим вариантом написать программы по условиям, представленным в таблице ниже. Использовать одномерный массив целых случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры. Представить результаты в окне **Отладчика**. Для одной из программ составить ***блок-схему*** *а*лгоритма.



1. Удалить пять первых нечетных элементов массива. Добавить в конец массива три новых нулевых элемента.

#include <iostream>

//подключаем библиотеку для работы с вводом/выводом

#include <ctime>

//директива для использования функции time

**using** **namespace** std;

//объявляем программе, что мы используем пространство имен std

**void** main() {

    setlocale(LC\_CTYPE, "Russian"); //добавляем возможность вывода на русском языке

**const** **int** N = 100; //определяем максимальный размер массива как 100

**int** sz, A[N], rmn = 0, rmx = 99, k, x = 1; //объявляем переменные

    cout << "Введите размер массива: ";

    cin >> sz;

    cout << "Массив: ";

    srand((**unsigned**)time(**NULL**)); //инициализирует генератор случайных чисел на основе текущего времени

**for** (**int** i = 0; i < sz; i++) {

        //заполняем массив с помощью оператора for

        A[i] = (**int**)(((**double**)rand() / (**double**)RAND\_MAX) \* (rmx - rmn) + rmn); //генерирует случайное число между rmn и rmx

        cout << A[i] << " "; //поочередно выводим элементы массива

    }

**for** (**int** i = 0; i < sz; i++) {

    //добавляем оператор for для проверки каждого числа в массиве

**if** ((A[i] % 2 != 0) && (x <= 5)) {

            //ищем первые 5 нечетных элементов

**for** (k = i; k <= sz; k++) {

                //удаляем нечетный элемент путем переопределения порядка в массиве

                A[k] = A[k + 1];

            }

            sz--; //уменьшаем размер вычисления для массива

            x++; //обозначаем найденный нечетный элемент

            i--; //так как мы переписывали массив, то нужно проверить следующее число под этим же номером

        }

    }

**for** (**int** i = 0; i < 3; i++) {

        //с помощью оператора for добавляем три 0 в конце массива

        A[sz] = 0;

        sz++;

    }

    cout << "\nМассив: ";

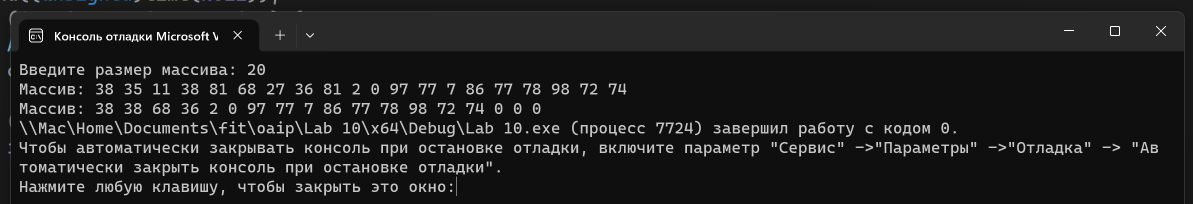
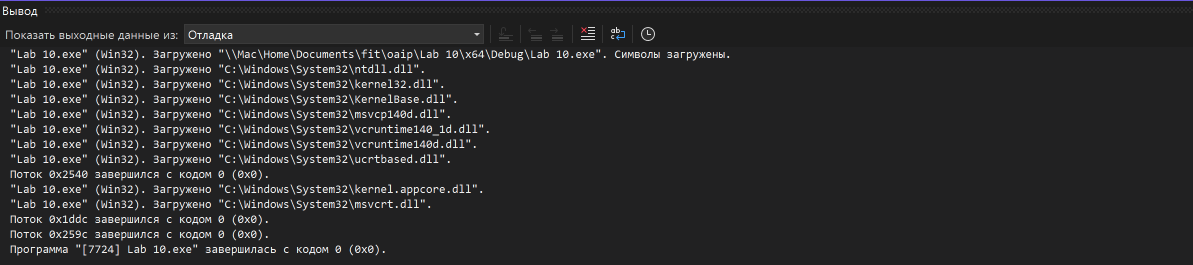
**for** (**int** i = 0; i < sz; i++) {

        cout << A[i] << " ";

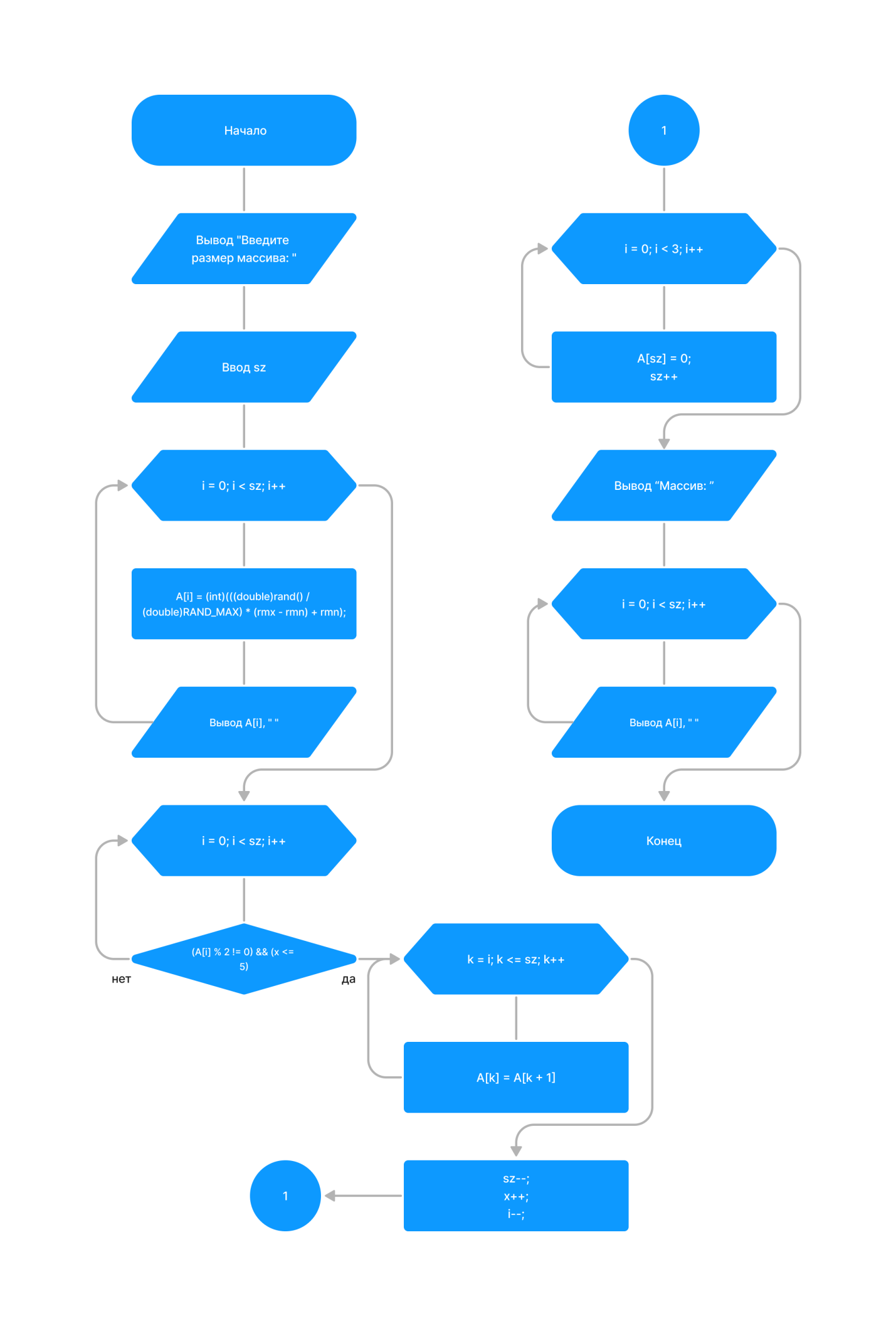
    }

    //выводим сообщение и измененный массив

}



Блок-схема:



1. Задан массив из **k** символов. Удалить из него повторные вхождения каждого символа.

#include <iostream>

//подключаем библиотеку для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы используем пространство имен std

**void** main(){

    setlocale(LC\_CTYPE, "Russian"); //добавляем возможность вывода на русском языке

**const** **int** N = 100; //определяем максимальный размер массива как 100

**int** sz = 12;

**char** A[] = { 'a', 'b', 'c', 'a', 'd', 'c', 'd', 'b', 'b', 'a', 'c', 's' }, B[N];

    cout << "Массив: ";

**for** (**int** i = 0; i < sz; i++) {

        cout << A[i] << " "; //поочередно выводим элементы массива

    }

**int**  d = 0, l = -1;

**for** (**int** i = 0; i < sz; i++) {

        //добавляем оператор for для сравнения всех символов

        d = 0;

**for** ( **int** c = i + 1; c < sz; c++) {

            //с помощью оператора for сравниваем последующие символы

**if** (A[i] != A[c]) {

                //если переменные не совпали, то добавляем 0

                d += 0;

            }

**else** {

                d++;

            //если нашли соответствие, то добавляем 1

            }

        }

**if** (d == 0) {

            l++;

            B[l] = A[i];

        //если соответствий не было, то добавляем символ в массив

        }

    }

    cout << "\nМассив: ";

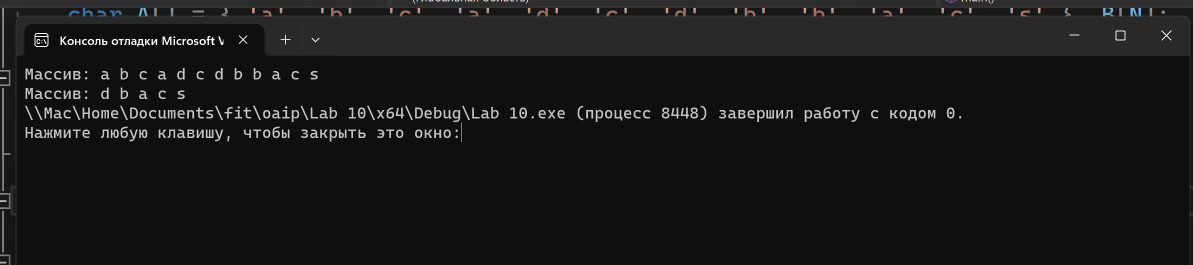
**for** (**int** i = 0; i <= l; i++) {

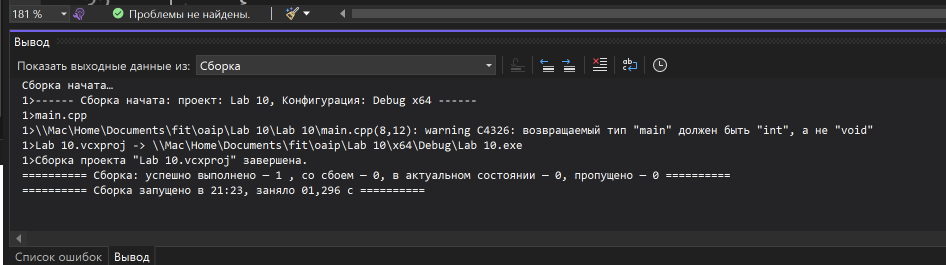
        cout << B[i] << " ";

    }

    //выводим сообщение и  массив

}





**Дополнительные задания**

1. В массиве M, размером k, много совпадающих элементов. Найти количество различных элементов в нем (не упорядочивая массив).

#include <iostream>

//подключаем библиотеку для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы используем пространство имен std

**void** main(){

    setlocale(LC\_CTYPE, "Russian"); //добавляем возможность вывода на русском языке

**int** sz = 12;

**char** A[] = { 'a', 'b', 'c', 'a', 'd', 'c', 'd', 'b', 'b', 'a', 'c', 's' };

    cout << "Массив: ";

**for** (**int** i = 0; i < sz; i++) {

        cout << A[i] << " "; //поочередно выводим элементы массива

    }

**int**  d = 0, l = 0;

**for** (**int** i = 0; i < sz; i++) {

        //добавляем оператор for для сравнения всех символов

        d = 0;

**for** ( **int** c = i + 1; c < sz; c++) {

            //с помощью оператора for сравниваем последующие символы

**if** (A[i] != A[c]) {

                //если переменные не совпали, то добавляем 0

                d += 0;

            }

**else** {

                d++;

            //если нашли соответствие, то добавляем 1

            }

        }

**if** (d == 0) {

            l++;

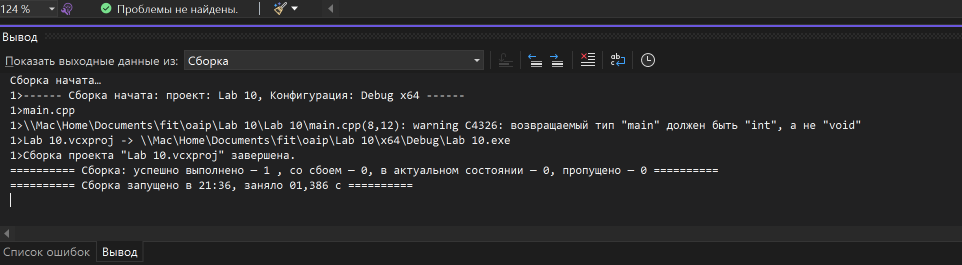
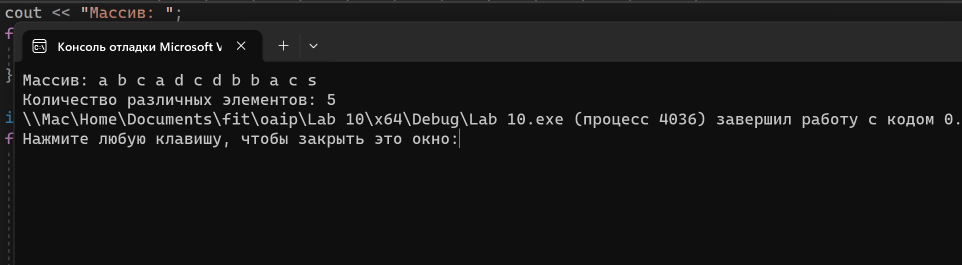
        //если соответствий не было, то дабавляем символ в массив

        }

    }

    cout << "\nКоличество различных элементов: " << l;

}



3) Найти в массиве наибольшее число подряд идущих одинаковых элементов.

#include <iostream>

//подключаем библиотеку для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы используем пространство имен std

**void** main(){

    setlocale(LC\_CTYPE, "Russian"); //добавляем возможность вывода на русском языке

**int** sz = 14;

**char** A[] = { 'a', 'a', 'c', 'a', 'd', 'c', 'c', 'c', 'd', 'b', 'b', 'a', 'c', 's' };

    cout << "Массив: ";

**for** (**int** i = 0; i < sz; i++) {

        cout << A[i] << " "; //поочередно выводим элементы массива

    }

**int**  d = 0, l = 0, max = 1;

**for** (**int** i = 0; i < sz; i++) {

        //добавляем оператор for для сравнения каждого с последующими

        d = i + 1;

        l = 1; //счетчик подряд идущих элементов

**for** ( **int** c = i; c < d; c++) {

            //с помощью оператора for сравниваем сравниваем рядом стоящие элементы, пока предыдущие совпадают

**if** (A[c] == A[d]) {

                //если переменные совпали, то увеличиваем счетчик и диапозон сравнения

                d++;

                l++;

            }

**if** (l > max) {

                max = l; //присваиваем максимальное значение подряд идущих одинаковых элементов

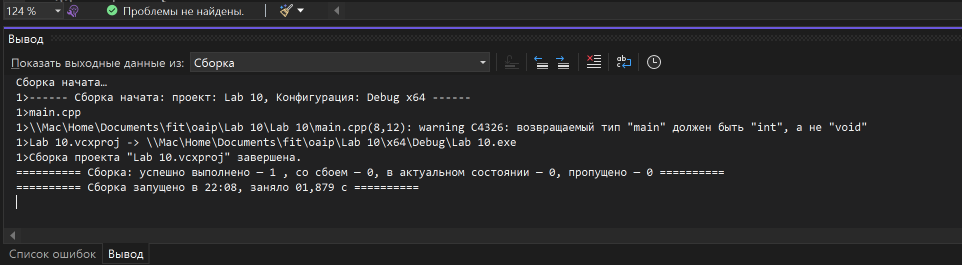
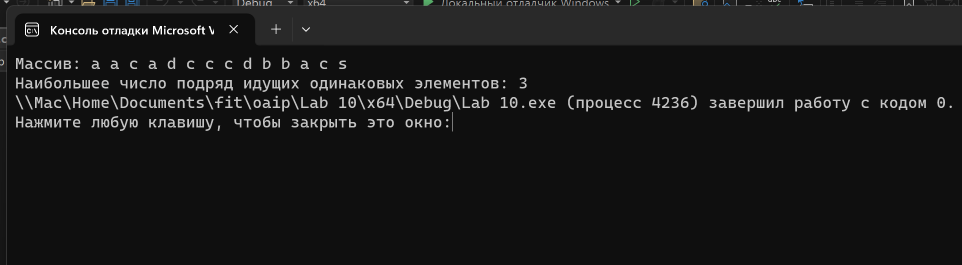
            }

        }

    }

    cout << "\nНаибольшее число подряд идущих одинаковых элементов: " << max;

}



2) Подсчитать количество пар соседних элементов массива с одинаковыми значениями.

#include <iostream>

//подключаем библиотеку для работы с вводом/выводом

**using** **namespace** std;

//объявляем программе, что мы используем пространство имен std

**void** main(){

    setlocale(LC\_CTYPE, "Russian"); //добавляем возможность вывода на русском языке

**int** sz = 14;

**char** A[] = { 'a', 'a', 'c', 'a', 'd', 'c', 'c', 'c', 'd', 'b', 'b', 'a', 'c', 's' };

    cout << "Массив: ";

**for** (**int** i = 0; i < sz; i++) {

        cout << A[i] << " "; //поочередно выводим элементы массива

    }

**int**  d = 0, l = 0;

**for** (**int** i = 0; i < sz; i++) {

        //добавляем оператор for для сравнения каждого с последующими

**if** (A[i] == A[i+1]) {

            //если переменные совпали, то увеличиваем счетчик

            l++;

        }

    }

    cout << "\nКоличество пар соседних элементов массива с одинаковыми значениями: " << l;

}

