Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

**«ИЗУЧЕНИЕ ОДНОМЕРНЫХ МАССИВ И СТРОК»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРИВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Балахнина Елизавета Ефимовна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы:** получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.
9. **Формулировка задания (Вариант:2)**

1. Написать программу, выполняющую поиск двух элементов массива, таких, что их разность по модулю минимальна.

2. Написать программу, определяющую, какое из слов встречается в заданной строке чаще всего.

3.Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением длины строки)

1. **Описание алгоритма**

***Задание № 1***

1. Через константу задать массиву максимальный размер.
2. Ввести размер и элементы массива
3. Первым шагом программа находит разницу между вторым и первым элементом массива.
4. Далее идет цикл, который проходится по всем элементам массива кроме последнего.
5. Для каждой пары соседних элементов находится разность. В случае, если найденная разница меньше первичного значения, программа сохраняет полученную разницу как первоначальную.
6. Вывести элементы с минимальной разницей по модулю и найденную минимальную разницу

***Задание № 2***

Данный код представляет собой программу, которая находит часто встречающееся слово.

1. Сначала пользователю предлагается ввести строку. Введенная строка сохраняется в переменной str. Потом объявляются необходимые переменные.
2. Затем начинается цикл for, который проходит по каждому символу введенной строки.
3. После окончания первого цикла, начинается второй цикл for, который проходит по каждому слову в массиве words.
4. Внутри второго цикла инициализируется переменная count\_2 для подсчета количества повторений текущего слова.
5. Затем запускается третий цикл for, который сравнивает слово в массиве с текущим словом.
6. Если текущее слово равно слову из массива words, то увеличиваем счетчик count\_2 на 1.
7. После окончания второго цикла проверяется, является ли значение count\_2 больше значения count\_1 или равно ему и значение res пустое.
8. Если условие выполняется, то сохраняем текущее слово в переменную res и обновляем значение count\_1 равным count\_2.
9. После окончания второго цикла проверяется, является ли значение res равным последнему слову из массива words, чтобы определить, было ли найдено наиболее часто повторяющееся слово.
10. После проверки условий выводится ‘Наиболее частое слово: ’ или ‘Нет повторяющихся слов: ’.

**4. Схема алгоритма с комментариями**

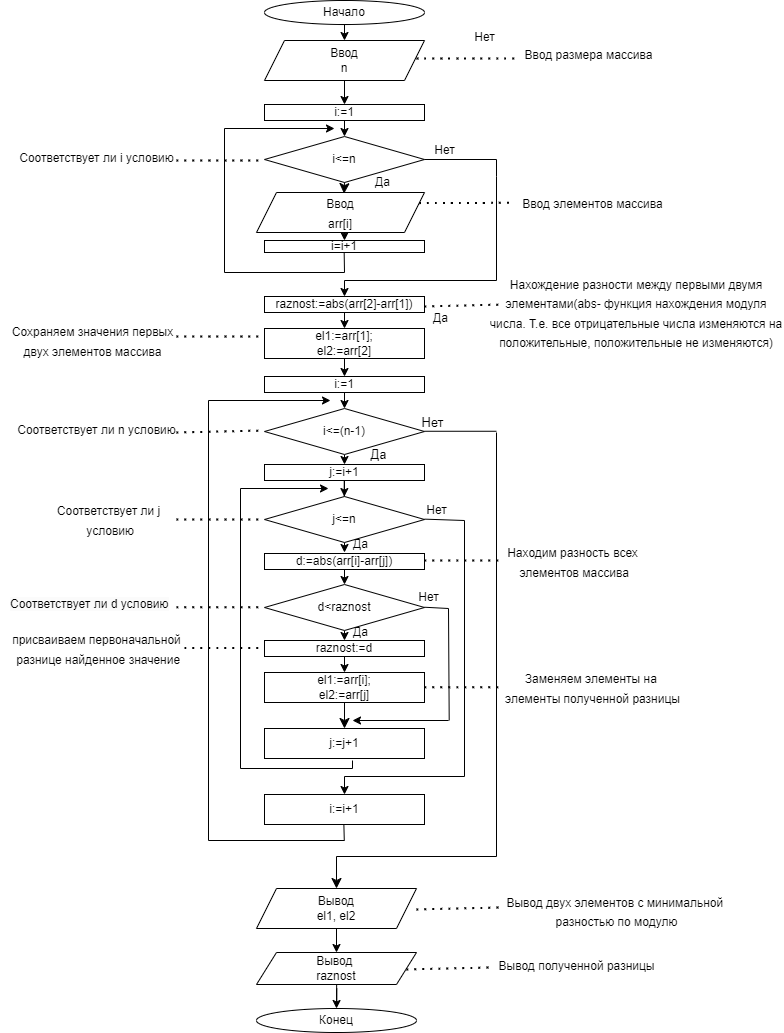


Рис 1. Схема алгоритма (задание 1) с комментариями.

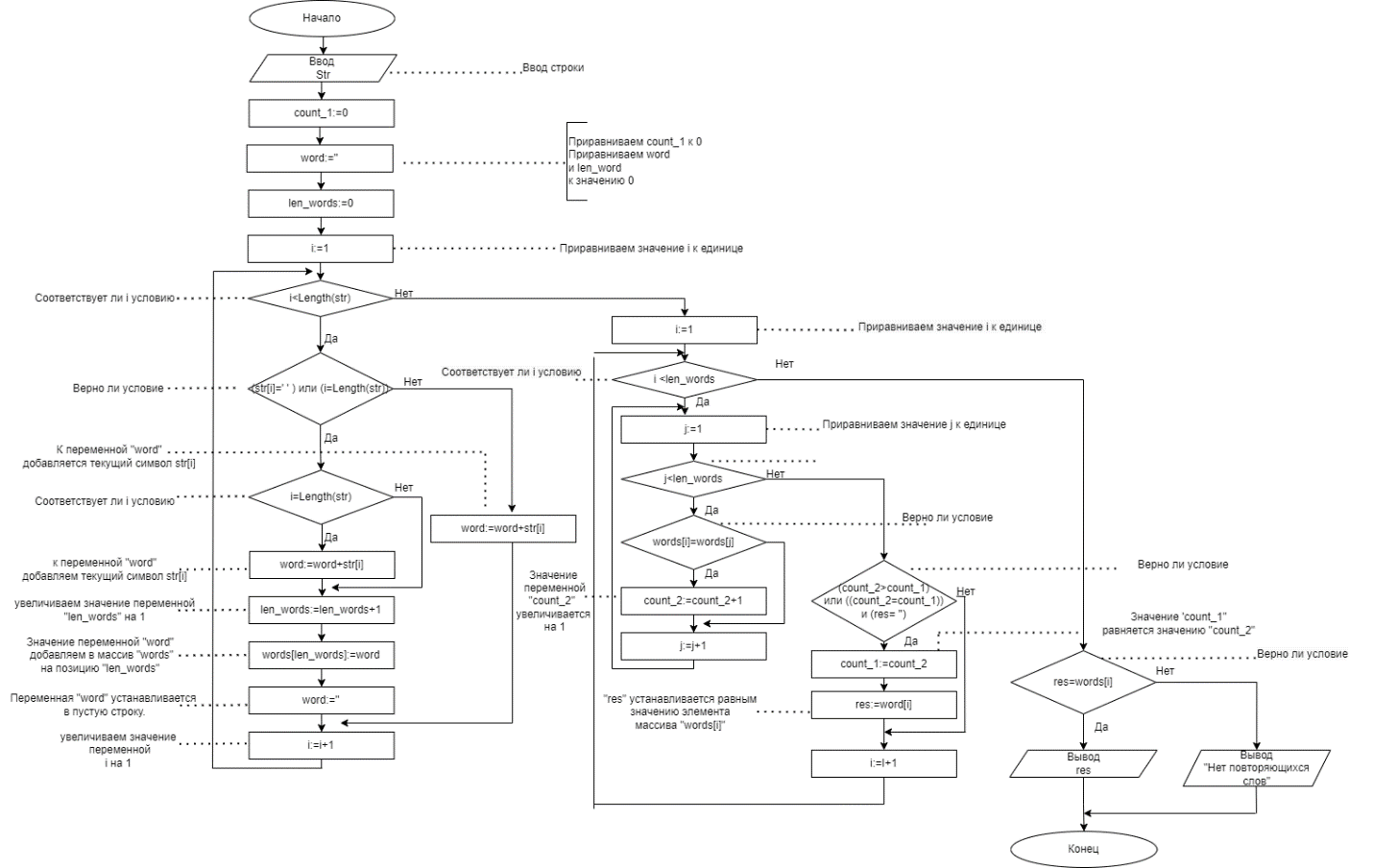


Рис 2. Схема алгоритма (задание 2) с комментариями.

1. **Код программы**

***Задача №1***

**program** z1;

**const**

max = 100;

**var**

arr: **array**[1..max] **of** Integer;

n, i, j, raznost, d, el1, el2: Integer;

**begin**

writeln('Введите размер массива (не более 100): ');

readln(n);

writeln('Введите элементы массива:');

**for** i := 1 **to** n **do**

readln(arr[i]);

raznost := abs(arr[2] - arr[1]);

el1 := arr[1];

el2 := arr[2];

**for** i := 1 **to** n - 1 **do**

**begin**

**for** j := i + 1 **to** n **do**

**begin**

d := abs(arr[i] - arr[j]);

**if** d < raznost **then**

**begin**

raznost := d;

el1 := arr[i];

el2 := arr[j];

**end**;

**end**;

**end**;

writeln('Два элемента с минимальной разностью по модулю: ', el1, ' и ', el2);

writeln('Разность: ', raznost);

**end**.

***Задача № 2***

**program** z2;

**var**

i, j, count\_1, count\_2: integer;

res, str, word: string;

words: **array**[1..100] **of** string;

len\_words: integer;

**begin**

writeln('Введите строку:');

readln(str);

count\_1 := 0;

word := '';

len\_words := 0;

**for** i := 1 **to** Length(str) **do**

**begin**

**if** (str[i] = ' ') **or** (i = Length(str)) **then**

**begin**

**if** i = Length(str) **then**

word := word + str[i];

len\_words := len\_words + 1;

words[len\_words] := word;

word := '';

**end**

**else**

word := word + str[i];

**end**;

**for** i := 1 **to** len\_words **do**

**begin**

count\_2 := 0;

**for** j := 1 **to** len\_words **do**

**if** words[i] = words[j] **then**

count\_2 := count\_2 + 1;

**if** (count\_2 > count\_1) **or** ((count\_2 = count\_1) **and** (res = '')) **then**

**begin**

count\_1 := count\_2;

res := words[i];

**end**;

**end**;

**if** res = words[i] **then**

writeln('Наиболее частое слово: ', res)

**else**

writeln('Нет повторяющихся слов');

**end**.

1. **Результат выполнения программы**

***Задача № 1***

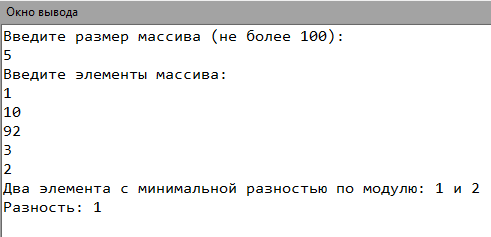
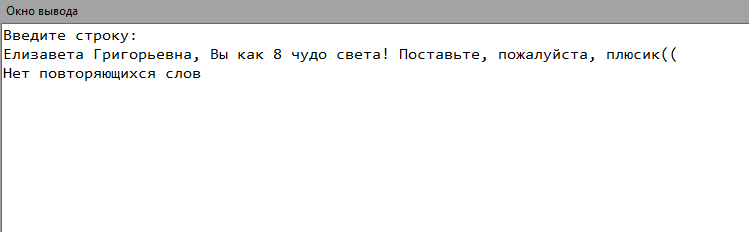


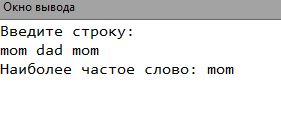
Рис.3. Результат выполнения программы (задание 1)

***Задача № 2***

******

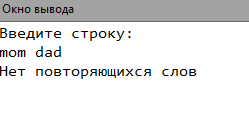
***(При условии ввода в строку слов на русском языке без повтора)***

Рис.4. Результат выполнения программы (задание 2)

******

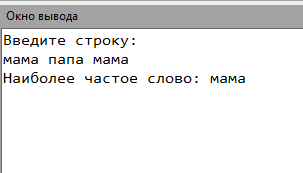
(***При условии ввода в строку слов на английском языке с повтором)***

Рис.5. Результат выполнения программы (задание 2)



***(При условии ввода в строку слов на английском языке без повтора)***

Рис.6. Результат выполнения программы (задание 2)



***(При условии ввода в строку слов на русском языке с повтором)***

Рис.7. Результат выполнения программы (задание 2)

**Вывод**

В процессе выполнения домашней контрольной работы, мы узнали множество новых операций на языке Паскаль, как например, функция «abs»

В языке программирования Паскаль (Pascal), функция abs используется для нахождения абсолютного значения числа. Абсолютное значение числа модуль числа, то есть оно всегда будет положительным. Функция abs принимает один аргумент — число, для которого нужно найти абсолютное значение, и возвращает положительное значение данного числа. Например, abs (-5) вернет 5, так как абсолютное значение числа -5 равно 5.

К тому же, мы столкнулись с такой проблемой, как запрет на использование стандартных функций для работы со строками (за исключением длины строки). Чтобы продолжить работу, нам пришлось использовать только функцию Length(s).

Length(s) в Pascal является встроенной функцией, которая возвращает длину массива s. Эта функция может использоваться для вывода количества элементов в массиве или для работы с циклами, чтобы обратиться к каждому элементу массива. Результатом функции будет целое число, соответствующее количеству элементов в массиве s. Только изучив аналоги системных операций, мы продолжили дальнейшее написание функции.

В процессе выполнения работы, мы познакомились с сайтом diagrams.net. Diagrams.net - это бесплатный онлайн-инструмент для создания диаграмм, схем, графиков и других визуальных элементов. Он представляет удобный интерфейс с возможностью создания различных типов диаграмм, таких как блок-схемы, организационные системы и многие другие.