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 МАССИВОВ И СТРОК»**

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

Выполнил: студент учебной группы

ИСПк-203-52-00

Шутова Юлиана Александровна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

**1.Цель работы:** получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками с частным случаем одномерных массивов.

**2. Формулировка задания (Вариант:25)**

1. Написать программу, определяющую является ли заданный массив монотонным.

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 функции определения длины строки.)

**3. Описание алгоритма**

**Задание 1**

Данный код представляет программу, которая определяет, является ли введенный пользователем массив монотонным.

1.На первых строках мы вводим размер массива и элементы массив

2. Затем устанавливаются переменные "n" и "d" в значение True.

3.После этого начинается следующий цикл, который выполняется от 1 до "b-1".

4. В цикле проверяется условие "если текущий элемент не равен предыдущему", то выполняется следующий блок кода. В этом блоке кода проверяется, больше ли текущий элемент предыдущего. Если это так, то переменная "d" устанавливается в значение False. Затем проверяется, меньше ли текущий элемент предыдущего. Если это так, то переменная "n" устанавливается в значение False.

5. После этого проверяется условие "если текущий элемент равен предыдущему", то выполняется следующий блок кода. В этом блоке кода также проверяется, больше ли текущий элемент предыдущего. Если это так, то переменная "d" устанавливается в значение False. Затем проверяется, меньше ли текущий элемент предыдущего. Если это так, то переменная "n" устанавливается в значение False.

6. После проверки всех условий цикла проверяются значения переменных "n" и "d". Если "n" равно True, то выводится сообщение "Массив является монотонным". Если "d" равно True, то выводится сообщение "Массив является монотонным.". В противном случае выводится сообщение "Массив не является монотонным.".

**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 описанием**

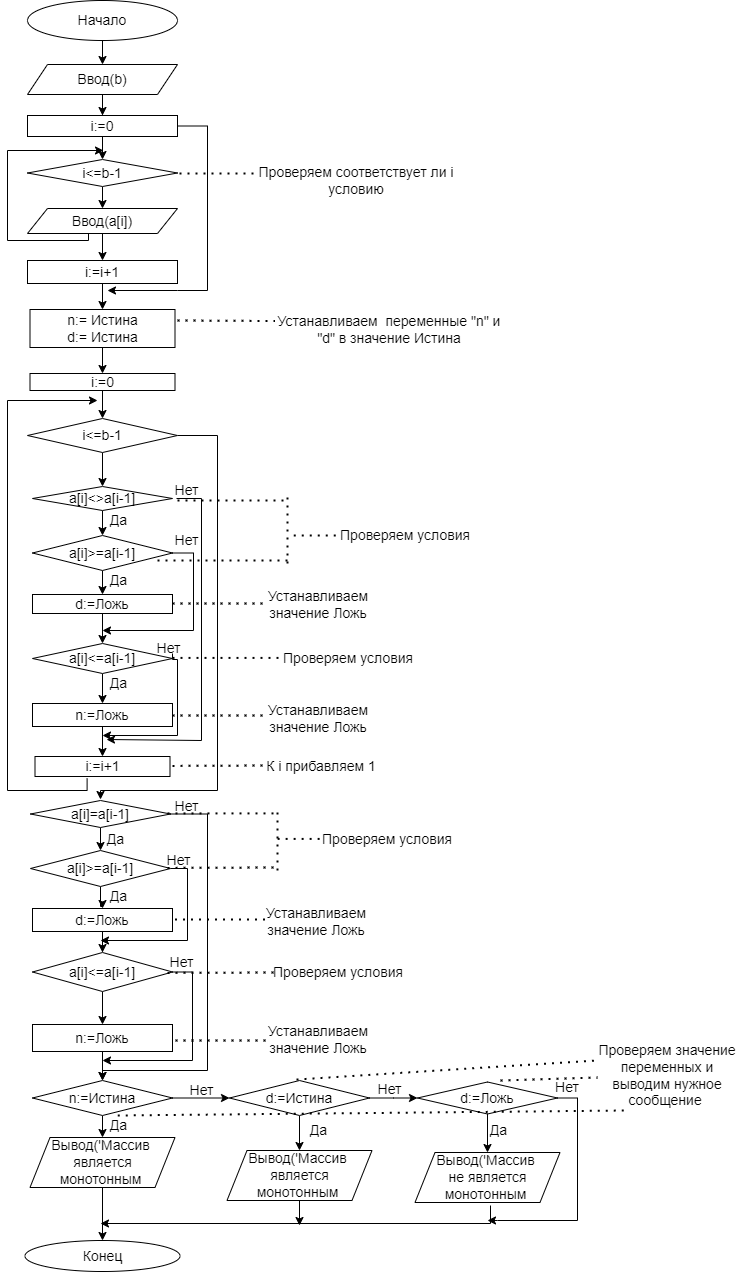
****

Рисунок 1 - Схема алгоритма с комментариями

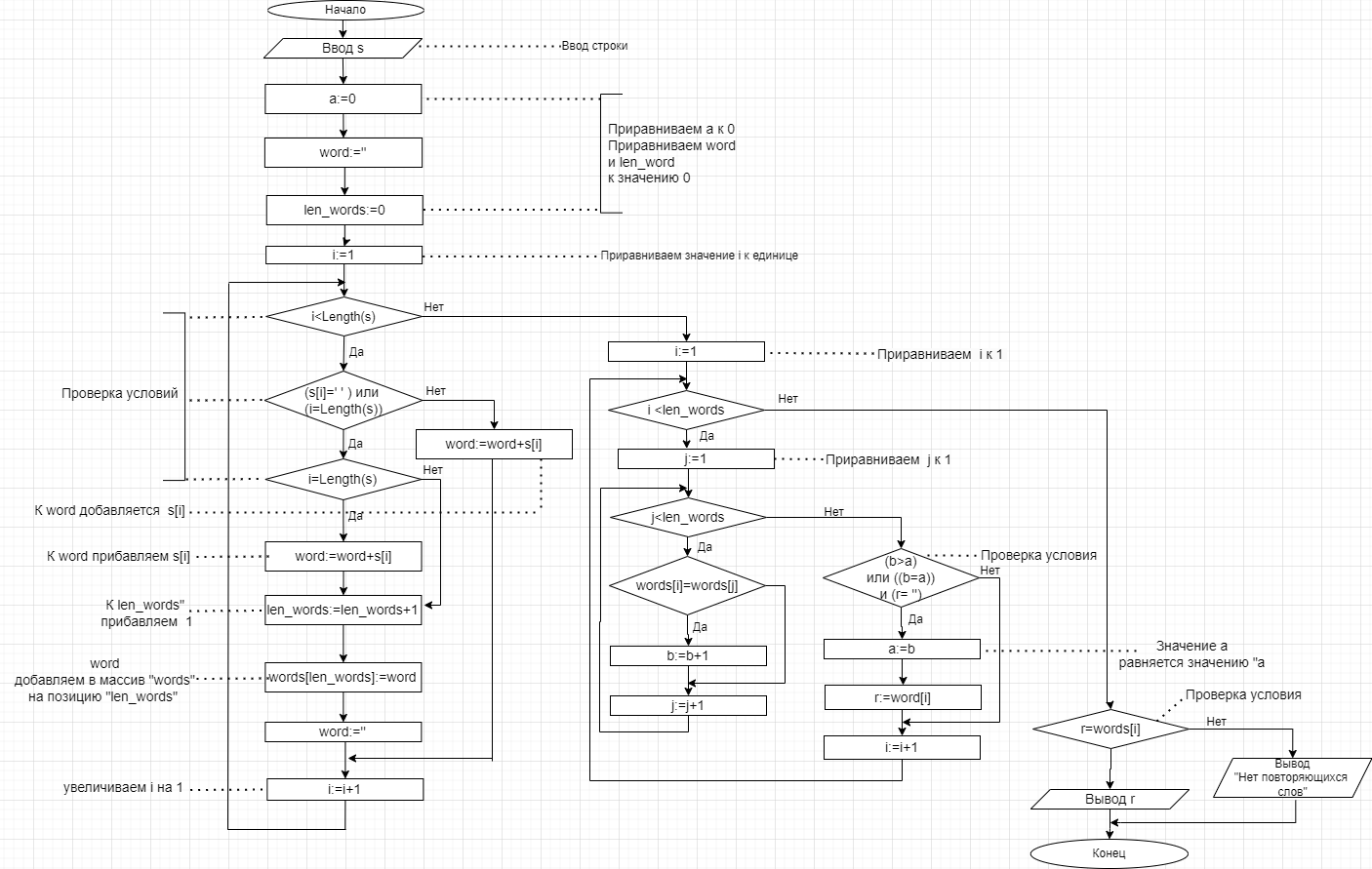
****

Рисунок 2 – Схема алгоритма с комментариями

**5. Код программы**

**Задание 1**

**program** z1;

**var**

a: **array of** Integer;

i,b: Integer;

n, d: Boolean;

**begin**

write('Введите размер массива: ');

read(b);

setLength(a,b);

write('Введите элементы массива: ');

**for** i := 0 **to** b-1 **do**

read(a[i]);

n:= True;

d:= True;

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

**if** (a[i] <> a[i - 1]) **then**

**begin**

**if** a[i] >= a[i - 1] **then**

d:= False;

**if** a[i] <= a[i - 1] **then**

n := False;

**end**;

**if** (a[i] = a[i - 1]) **then**

**begin**

**if** a[i] >= a[i - 1] **then**

d:= False;

**if** a[i] <= a[i - 1] **then**

n:= False;

**end**;

**if** n **then**

writeln('Массив является монотонным')

**else if** d **then**

writeln('Массив является монотонным.')

**else**

writeln('Массив не является монотонным.');

**end**.

**Задание 2**

**program** z2;

**var**

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

res, str, word: string;

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

length\_words: integer;

**begin**

writeln('Введите строку:');

readln(str);

count\_1 := 0;

word := '';

length\_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];

length\_words := length\_words + 1;

words[length\_words] := word;

word := '';

**end**

**else**

word := word + str[i];

**end**;

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

**begin**

count\_2 := 0;

**for** j := 1 **to** length\_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**.

**6. Результат выполнения программы**

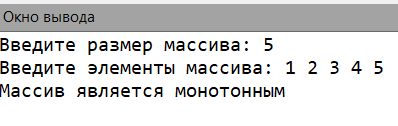
****

Рисунок 3 – Результат выполнения программы 1

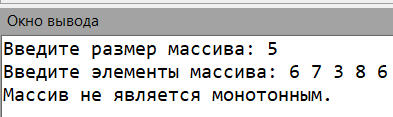
****

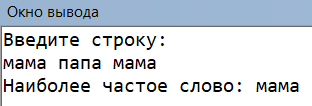
Рисунок 4 – Результат выполнения программы 1

Рисунок 5 – Результат выполнения программы 2

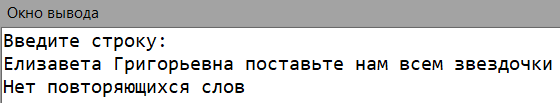
****

Рисунок 6 – Результат выполнения программы 2

**7. Вывод**

Данная домашняя контрольная работа была написана на языке программирования Pascal. В данной работе нам требовалось написать программу для определения монотонности массива и нахождения часто повторяющегося слова в строке.

Во время выполнения домашней контрольной мы изучили работу с массивами и строками.

Для выполнения задания 2 нам запрещалось использовать стандартные функции. Это усложнило написание кода, но разрешалось использовать функцию определения длины строки. В дальнейшем проблем в написании программ не возникало.

В процессе выполнения работы, для создания схем алгоритма для задач мы использовали уже знакомую нам программу draw.io. Draw.io – это инструмент для создания диаграмм, блок-схем, интеллект-карт и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя. Сервис позволяет отслеживать и восстанавливать изменения готовых продуктов, импортировать и экспортировать в PDF, PNG, XML, VSDX, HTML, а также автоматически публиковать и делиться работами. Инструмент работает с GoogleДиск, GoogleWorkspace и Dropbox.