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 ПРОГРАММИРОВАНИЯ**»**

Выполнил: студент учебной группы

ИСПк-208-52-00

ФИО: Потапов Максим Денисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы

Изучить базовую структуру организации программы и основные конструкции языка программирования Pascal.

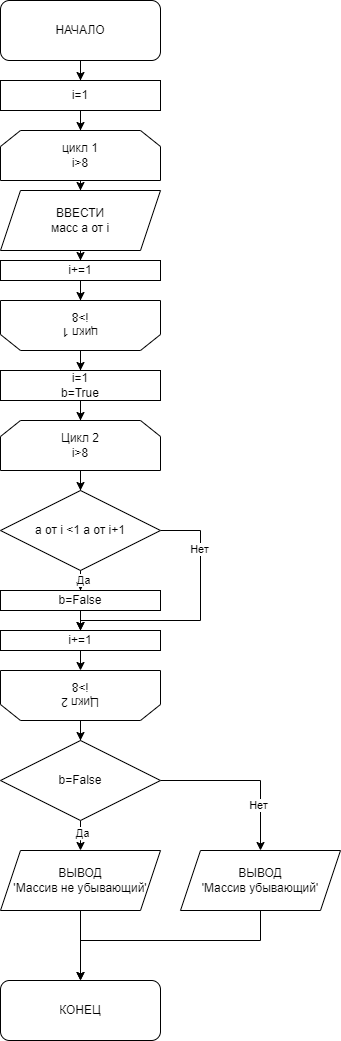
1. Формулировка заданий
2. Написать программу, определяющую является ли заданный массив убывающим.
3. Написать программу, определяющую количество уникальных подстрок из трёх символов, входящих в заданную строку.
4. Описание алгоритма №1

Алгоритм 1-ый:

1. Определяется тип переменных.
2. Вводится массив. .
3. Запускается цикл, который сравнивает каждый элемент массива со следующим. Если следующая переменная больше предыдущей, то проверяющей переменной задаётся значение False.
4. Выводится ответ зависимый от проверяющей переменной.

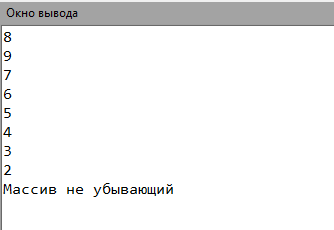
Алгоритм 2-ой:

1. Определяется тип переменных.
2. Вводится строка.
3. Запускается цикл, который переписывает в строковый массив все не повторяющиеся слова из трёх букв из строки.
4. Вычисляется длина строкового массива.
5. Длина массива выводится на экран.
6. Схема первого алгоритма



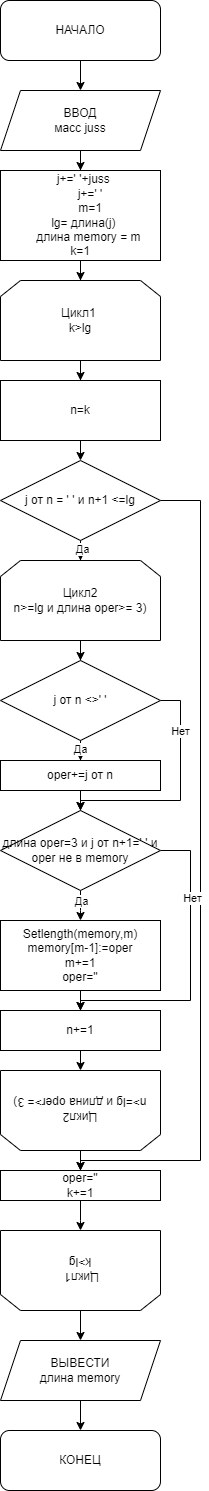
**Рисунок 1 Первый алгоритм**

1. Код программы для первой задачи
2. **var**
3. a:**array**[1..8] **of** integer;
4. i:integer;
5. b:boolean;
7. **begin**
8. **for** i:=1 **to** 8 **do**
9. readln(a[i]);
10. i:=1;
11. b:=True;
12. **while** i<8 **do begin**
13. **if** a[i] <1 a[i+1] **then**
14. b:=False;
15. i+=1;
16. **end**;
17. **if** b = False **then**
18. writeln('Массив не убывающий')
19. **else**
20. writeln('Массив убывающий');
21. **end**.
22. Результат выполнения программы для первой задачи



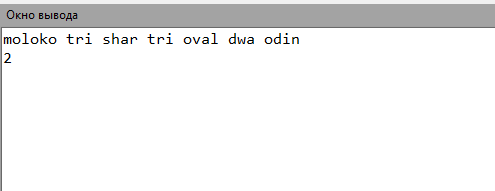
**Рисунок 2 Результат**

1. Схема второго алгоритма



**Рисунок 3 Второй алгоритм**

1. Код программы для второй задачи
2. **var**
3. juss,j,oper:string;
4. memory:**array of** string;
5. n,m,mn,k,lg,count:integer;
6. **begin**
7. j:=' ';
8. readln(juss);
9. j+=' '+juss;
10. j+=' ';
11. m:=1;
12. lg:=Length(j);
13. Setlength(memory,m);
14. **for** k:= 1 **to** lg **do begin**
15. n:=k;
16. **if** (j[n] = ' ') **and** (n+1 <=lg) **then begin**
17. **while** (n<lg) **and** (length(oper)< 3) **do begin**
18. **if** j[n]<>' ' **then**
19. oper+=j[n];
20. **if** (length(oper)=3) **and** (j[n+1]=' ') **and** (oper **not in** memory) **then begin**
21. Setlength(memory,m);
22. memory[m-1]:=oper;
23. m+=1;
24. oper:='';
25. **end**;
26. n+=1;
27. **end**;
29. **end**;
30. oper:='';
31. **end**;
32. mn:=Length(memory);
33. writeln(mn)
34. **end**.
35. Результат выполнения программы для второй задачи



**Рисунок 4 Результат**

1. Вывод

В данном отчете предоставлены данные и решения задач, которые были даны в домашней контрольной работе для написания и решения на языке программирования Pascal. Для выполнения были применены знания, которые давались на лекциях и практиках по дисциплине «Основы алгоритмизации и программирования. Так же, помогли базовые и более углубленные знания в математике, в частности, в алгебре.

При выполнении задания возникли небольшие трудности со второй задачей. Для её решения прошлось возвращаться на тему назад и перечитывать материал про циклы и их разновидности.