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»**

Выполнила: студентка учебной группы

ИСПк-203-52-00

Носкова Анастасия Владимировна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

**Задача 1 (Вариант 16)**

Написать программу, определяющую является ли заданный массив возрастающим.

**Описание алгоритма**

1. Объявим константу m равную 5;
2. Инициализируем a как массив целых чисел размером m, i как целые числа, n как значения True или False;
3. С помощью цикла For вводим массив a;
4. С помощью второго цикла For проверяем каждый элемент массивы на выполнение условия a[i]>a[i+1], если это условие выполняется, то переменной n присваивается значение False и цикл прекращается, иначе n присваивается значение True;
5. После выхода из цикла идет проверка условия True, если условие выполняется, то на экран выводится «Массив является возрастающим», иначе «Массив не является возрастающим».

**Схема алгоритма**

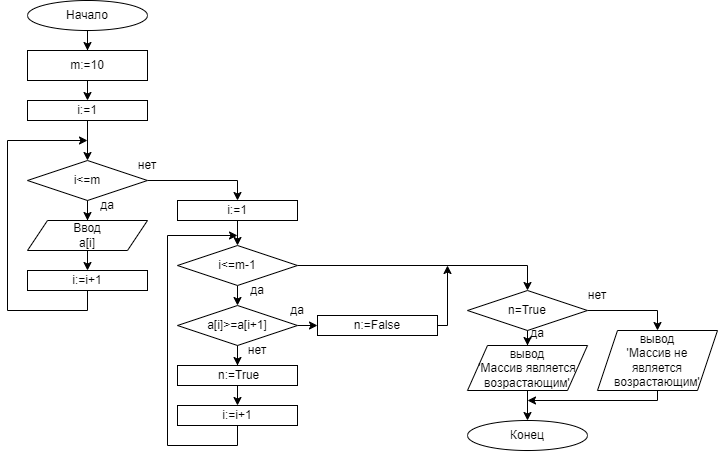


Рис.1 - Схема алгоритма 1 задания

**Код решения программы**

const m=5;

var a: array [1..m] of integer;

i: integer;

n: boolean;

begin

writeln('Введите массив: ');

for i:=1 to m do

read(a[i]);

for i:=1 to m-1 do begin

if a[i]>a[i+1] then begin

n:= false;

break;

end

else

n:= true;

end;

if n then writeln ('Массив является возрастающим')

else writeln('Массив не является возрастающим')

end.

**Результат выполнения программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 2 3 4 5 | Массив является возрастающим |
| -1000 2 140 19 200 | Массив не является возрастающим |
| 9 0 12 13 14 | Массив не является возрастающим |
| -200 -100 0 100 200 | Массив является возрастающим |

**Задача 2 (вариант 16)**

Написать программу, определяющую количество уникальных подстрок из трех символов, входящих в заданную строку.

**Описание алгоритма**

1. Инициализируем s как строку, i и j, служащие счетчиком, dl(длина строки) и z (подсчет количества уникальных подстрок) как целые числа, n как значения True или False;
2. Ввод строки;
3. Присваиваем переменной z значение 0, а переменной dl длину данной строки;
4. В цикле For j присваивается значение 0, n – False;
5. В нем запускается цикл while, который будет выполняться пока j<dl-3;
6. Внутри цикла while проверяются условия: если (i+1=j+1) and (i+2=j+2) and (i+3=j+3), то j увеличивается на 1. Если s[dl-2]=s[i+1]) and (s[dl-1]=s[i+2]) and (s[dl]=s[i+3]) and (i+1<>dl-2) and (i+2<>dl-1) and (i+3<>dl), то n становится true. Если (s[i+1]=s[j+1]) and (s[i+2]=s[j+2]) and (s[i+3]=s[j+3]), то n становится true;
7. Значение j увеличивается на 1;
8. Если n равен False, то z увеличивается на 1;
9. После завершения цикла for выводится количество уникальных подстрок из трех символов: z

**Схема алгоритма**

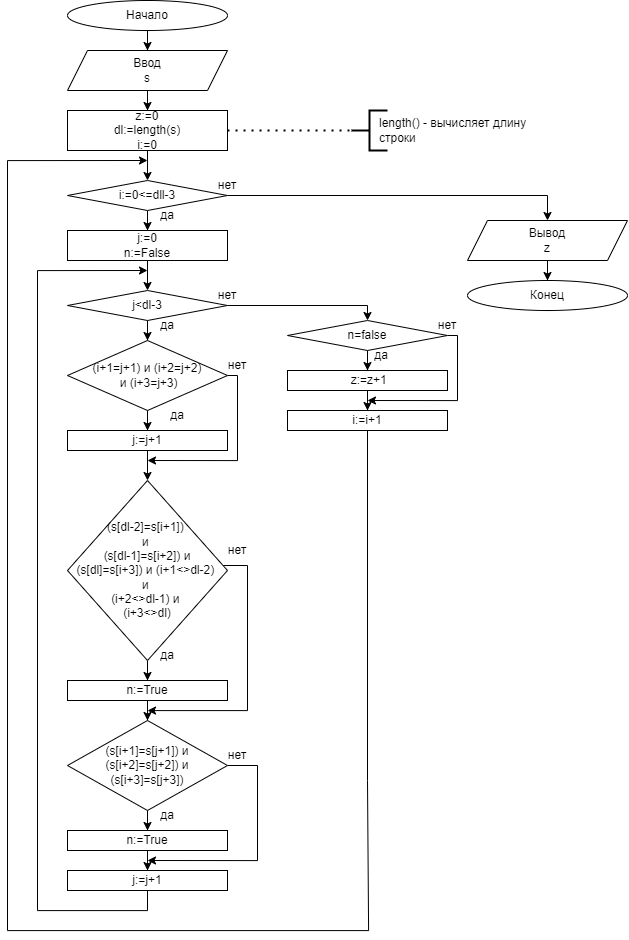


Рис.2 – Схема алгоритма 2 задания

**Код решения программы**

program func2;

var

s: string;

i, dl, j, z: integer;

n: Boolean;

begin

writeln('Введите строку');

readln(s);

z:=0;

dl:=length(s);

for i:=0 to dl-3 do

begin

j:=0;

n:=false;

while j<dl-3 do

begin

if (i+1=j+1) and (i+2=j+2) and (i+3=j+3) then

j:=j+1;

if (s[dl-2]=s[i+1]) and (s[dl-1]=s[i+2])

and (s[dl]=s[i+3]) and (i+1<>dl-2) and (i+2<>dl-1) and (i+3<>dl) then

n:=true;

if (s[i+1]=s[j+1]) and (s[i+2]=s[j+2]) and (s[i+3]=s[j+3]) then

n:=true;

j:=j+1;

end;

if n=false then

z:=z+1;

end;

write('Количество уникальных подстрок из трёх символов: ', z)

end.

**Результат выполнения программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Привет | Количество уникальных подстрок из трёх символов: 4 |
| abcdddddnc | Количество уникальных подстрок из трёх символов: 5 |
| ппррииввееттт | Количество уникальных подстрок из трёх символов: 11 |
| красота природыыы | Количество уникальных подстрок из трёх символов: 15 |

**Вывод**

В результате выполнения работы возникли трудности, связанные с составлением условий в задании 2 на языке программирования Pascal.

В данной работе дополнительными источниками пользоваться не приходилось, так как одномерный массив трудностей не вызвал. Основная сложность заключалась в задании 2 связанной со строками, так как задача была понята неверно из-за чего находились иные значения от нужных, а именно количество слов с тремя буквами в строке, а не подстрок как требовалось.

Но методом проб и ошибок получилось достичь нужного результата.