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**

**<<Вычисление значения функции>>**

**ПО ДИСЦИПЛИНЕ <<Основы алгоритмизации и программирования>>**

Выполнил: студент учебной группы

ИСПк-101-51-00

Шертосбитов Михаил Романович

Преподаватель: Сергеева Елизавета Григорьевна

Киров

2024

Цель работы: Получить базовые навыки работы с одномерными массивами освоить принципы работы со строками как с частным случаем одномерных массивов

Задание Написать программу выполняющую поиск двух любых элементов заданного массива таких что их произведение не превосходит их суммы

**program** Project1;

**const**

maxelem = 100;

**var**

a: **array** [1..maxelem] **of** real; // Переменная для хранения массива.

i,j: integer; // Вспомогательная переменная для организации цикла.

n: integer; // Переменная для хранения значения количества элементов массива.

flag: boolean; // Логическая переменная для вывода информации о решении задачи.

**begin**

writeln('Введите значение количества элементов массива (не более 100):');

Readln(n);

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

**begin**

write('Введите элемент ',i,': ');

Readln(a[i]);

**end**;

writeln('Исходный массив:');

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

**begin**

Write(a[i]:5:1);

**if** i **mod** 10 = 0 **then**

Writeln;

**end**;

writeln;

flag:= false;

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

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

**begin**

// writeln('a[',i,']= ',a[i]:0:1,'; a[',j,']= ',a[j]:0:1,'; Sum = ',a[i] + a[j]:0:1,'; Mult = ',a[i]\*a[j]:0:1);

{Закомментаренная строчка для контроля правильности вычислений программы.

Если нужно проверить как работает программа, просто снимите комментирующие слеши "/"};

**if** a[i] + a[j] >= a[i]\*a[j] **then**

flag:= true;

**end**;

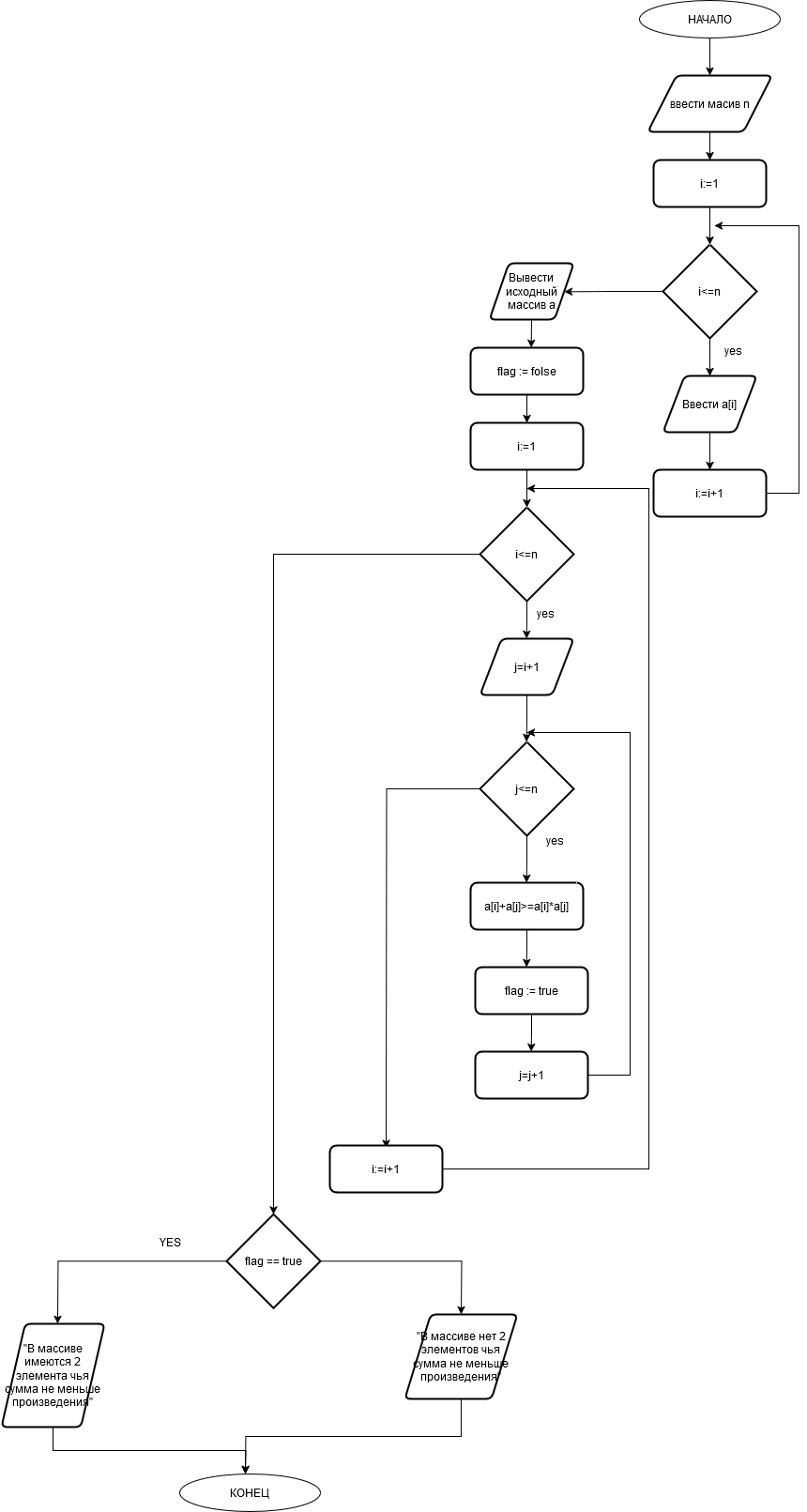
**if** flag **then**

writeln('В массиве имеются 2 элемента, чья сумма не меньше произведения.')

**else**

writeln('В массиве нет 2 элементов, чья сумма не меньше произведения.');

**end**.



2) написать программу вычисляющую вхождение одной строки в другую в качестве подстроки

**var**

a,b: string; // Переменные для хранения строк.

i,j: integer; // Вспомогательные переменные для организации циклов.

flag: boolean; // Логическая переменная для обнаружения несоответствия.

k: integer; // Переменная для организации счета.

**begin**

writeln('введи строку: ');

Readln(a);

writeln('введи входящею строку: ');

Readln(b);

k:= 0;

**for** i:= 1 **to** length(a) **do**

**begin**

flag:= false;

**for** j:= 1 **to** length(b) **do**

**if** a[i-1+j] <> b[j] **then**

**begin**

flag:= true;

**break**;

**end**;

**if not** flag **then**

inc(k);

**end**;

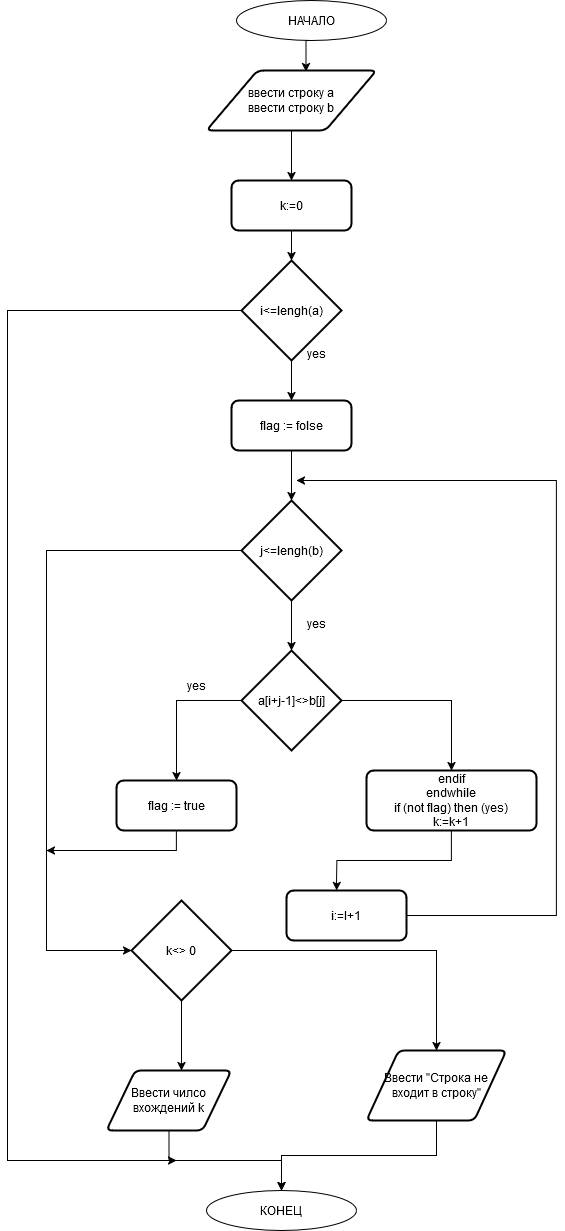
**if** k <> 0 **then**

writeln('число вхождений: ',k)

**else**

writeln('строка не входит в первую');

**end**.



YES

Вывод

Мы научились создавать одномерные массивы, задавать их размер и инициализировать элементы. Это позволило нам понять, как организовывать данные в памяти и как к ним обращаться.

Мы изучили, как строки в Pascal могут рассматриваться как массивы символов. Это дало возможность применять те же операции, что и с обычными массивами, к строкам, включая доступ к отдельным символам, изменение значений и перебор элементов.

В рамках работы мы реализовали алгоритм поиска вхождения одной строки в другую, что позволило закрепить навыки работы с массивами и строками. Мы использовали функции для поиска, сравнения и обработки строк, что является важной частью работы с текстовыми данными.

Мы реализовали логику подсчета количества вхождений одной строки в другую, что дало возможность понять, как можно использовать циклы и условия для обработки данных.

Лабораторная работа позволила нам получить практические навыки работы с одномерными массивами и строками в языке Pascal. Мы научились использовать массивы для хранения и обработки данных, а также поняли, как строки могут быть представлены как массивы символов.