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-205-52-00

Арбузов Иван Андреевич

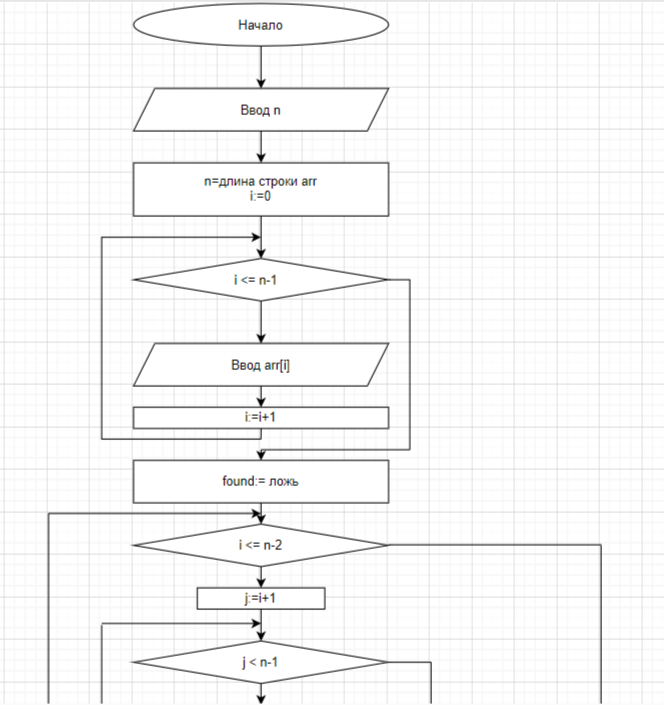
Преподаватель:

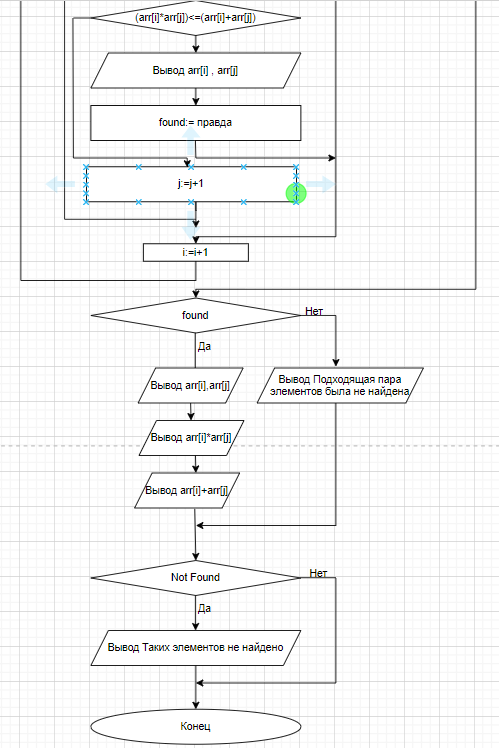
Сергеева Елизавета Григорьевна

Киров

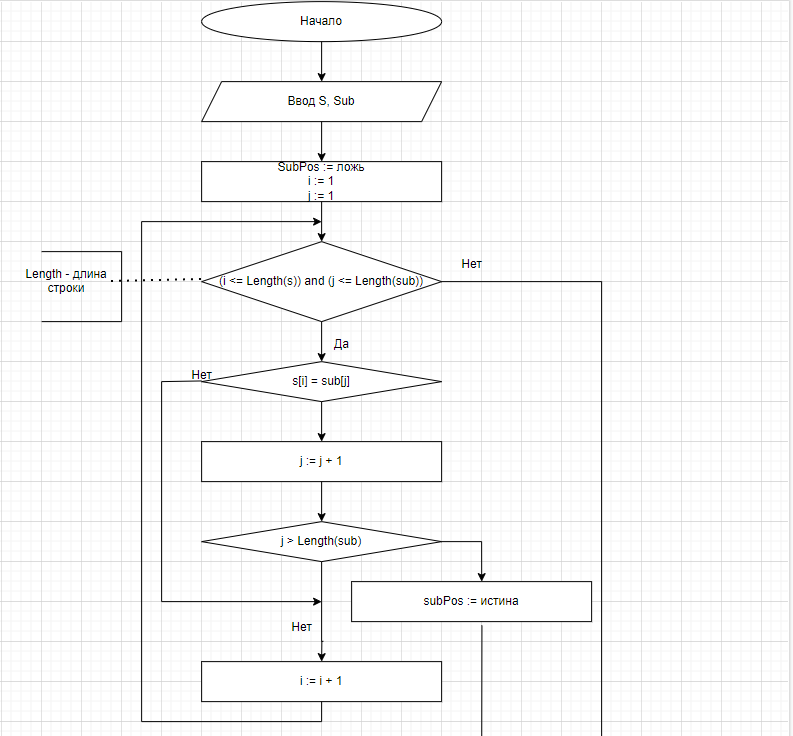
2023

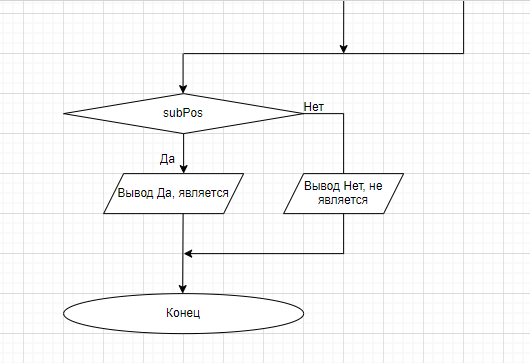
1. Цель работы: получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.
2. Вариант 20:
3. Написать программу, выполняющую поиск двух любых элементов заданного массива, таких, что их произведение не превосходит их суммы.
4. Написать программу, определяющую является ли заданная строка подпоследовательностью исходной строки.
5. Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длины строки).
6. Описание алгоритма:
7. Вводится Программа генерирует случайный массив из N элементов и выполняют поиск двух элементов, удовлетворяющих условию, что произведение элементов не превосходит их сумму. Если такие элементы найдены, они выводятся на экран. Если таких элементов нет, выводится соответствующее сообщение.
8. Программа сначала запрашивают у пользователя ввод исходной строки и подстроки, а затем проверяют, является ли подстрока подпоследовательностью исходной строки. Результат выводится на экран.
9. 1)





2)





1. Код программы:

**var**

arr: **array of** Integer;

n, i, j: Integer;

found: Boolean;

**begin**

Write('Введите количество элементов в массиве: ');

ReadLn(n);

SetLength(arr, n);

WriteLn('Введите элементы массива:');

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

**begin**

ReadLn(arr[i]);

**end**;

found := False;

**for** i := 0 **to** n-2 **do**

**begin**

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

**begin**

**if** (arr[i] \* arr[j]) <= (arr[i] + arr[j]) **then**

**begin**

WriteLn('Найдены элементы: ', arr[i], ' и ', arr[j]);

found := True;

**Break**;

**end**;

**end**;

**if** found **then**

**begin**

Writeln('Найдены элементы: ', arr[i], ' и ', arr[j]);

Writeln('Их произведение = ', arr[i] \* arr[j]);

Writeln('Их сумма = ', arr[i] + arr[j]);

**end**

**else**

Writeln('Подходящая пара элементов не найдена.');

**Break**;

**end**;

**if not** found **then**

WriteLn('Таких элементов не найдено');

**end**.

**2)**

**var**

s, sub: string;

i, j: integer;

subPos: boolean;

**begin**

Write('Введите исходную строку: ');

ReadLn(s);

Write('Введите подстроку: ');

ReadLn(sub);

subPos := False;

i := 1;

j := 1;

**while** (i <= Length(s)) **and** (j <= Length(sub)) **do**

**begin**

**if** s[i] = sub[j] **then**

**begin**

j := j + 1;

**if** j > Length(sub) **then**

**begin**

subPos := True;

**Break**;

**end**;

**end**;

i := i + 1;

**end**;

**if** subPos **then**

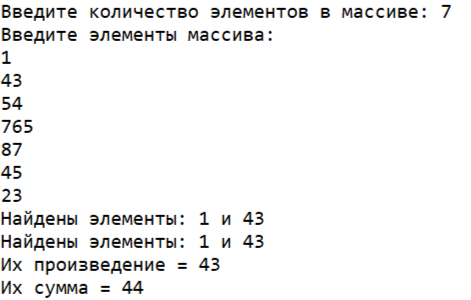
WriteLn('Подстрока является подпоследовательностью исходной строки.')

**else**

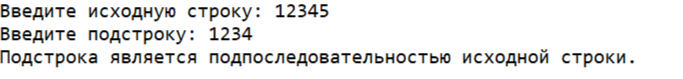
WriteLn('Подстрока не является подпоследовательностью исходной строки.');

**end**.









1. Вывод: В процессе выполнения контрольной работы мы получили базовые навыки работы с одномерными массивами, освоили принципы работы со строками как с частным случаем одномерных массивов. Мы узнали нового оператора как Break, который обычно его используют в связке с каким-либо условием. Если оно соблюдается или нет, то break останавливает цикл, а алгоритм продолжает выполнения дальнейших операторов.