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

Бочкарёва Виктория Дмитриевна

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.

**Формулировка задания**

Фамилия: Бочкарева

Вариант: 3

Задание:

1. Написать программу, определяющую является ли заданный массив возрастающим.
2. Написать программу, удаляющую из строки каждое второе вхождение заданной подстроки.
3. Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длины строки).

**Описание алгоритма**

Для решения первой задачи необходимо использовать цикл *for* для заполнения массива, а также для поверки данного массива на возрастание, конструкция *read()* для заполнения массива с клавиатуры. Для проверки на возрастание используется условный оператор *if...then…else*. В результате выполнения алгоритма будет проверяться больше ли следующий элемент массива, чем предыдущий, если это так, то переменной *k* будет присваиваться значение *false* (изначально *k = true*). Если *k = true*, то на экран выводится сообщение о том, что массив является возрастающим, с помощью конструкции *writeln()*, а в противном случае выводится сообщение о том, что массив не возрастающий.

Для решения второй задачи используются циклы *while* для подсчета вхождений заданной подстроки в данной строке, и условные операторы *if…then…else* для нахождения и пропуска подстроки, также для увеличения переменных используется процедура *Inc()*. Результатом выполнения программы является строка без каждого второго вхождения подстроки.

**Схема алгоритма**

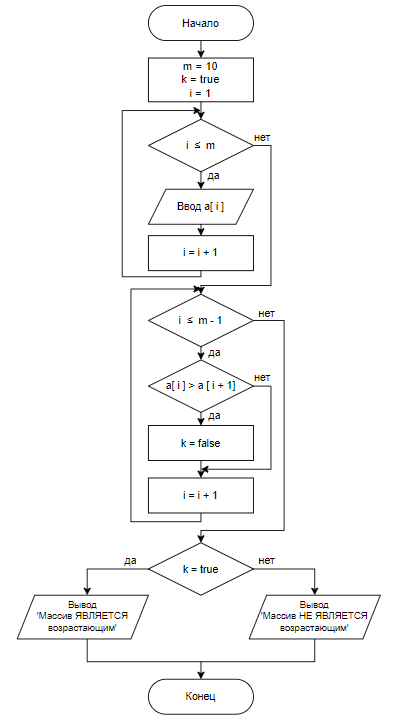
****

Рис. 1 – Схема алгоритма программы №1

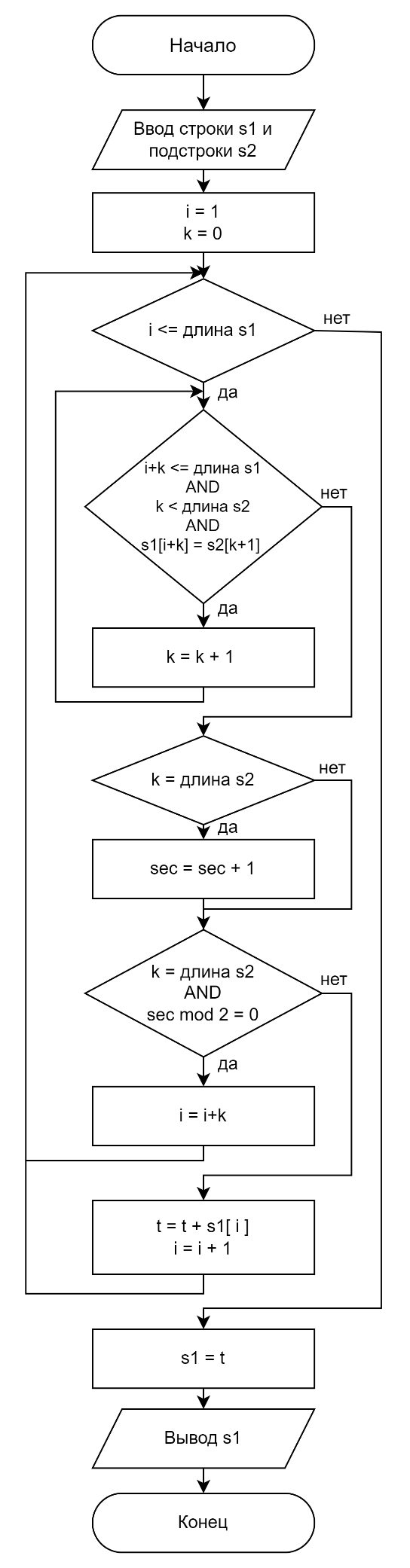


Рис. 2 – Схема алгоритма программы №2

**Код программы**

Код программы к заданию 1:

**program** dkr2\_1;

**const**

m = 10;

**var**

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

i: integer; k: boolean;

**begin**

k := true;

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** k := false;

**end**;

**if** k = true **then** writeln('Массив ЯВЛЯЕТСЯ возрастающим')

**else** writeln('Массив НЕ ЯВЛЯЕТСЯ возрастающим');

**end**.

Код программы к заданию 2:

**program** dkr2\_2;

**var**

s1, s2, t: string;

i, j, k, sec: integer;

**begin**

writeln('Введите строку: ');

readln(s1);

writeln('Введите подстроку: ');

readln(s2);

i:=1;

**while** i <= length(s1) **do**

**begin**

k:=0;

**while** ((i+k) <= length(s1)) **and** (k < length(s2)) **and** (s1[i+k] = s2[k+1]) **do** inc(k);

**if** k = length(s2) **then** inc(sec);

**if** (k = length(s2)) **and** (sec **mod** 2 = 0) **then** inc(i, k)

**else begin**

t+=s1[i];

inc(i);

**end**;

**end**;

s1:=t;

writeln(s1);

**end**.

**Результат выполнения программы**

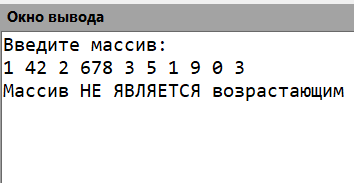
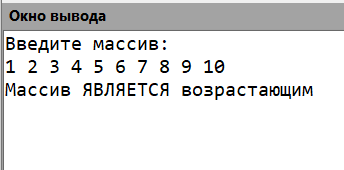


Рис. 3 – Результат выполнения программы №1

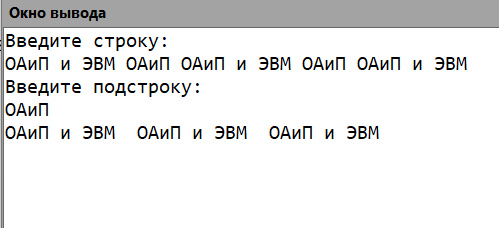


Рис. 4 – Результат выполнения программы №2

**Вывод**

В результате выполнения домашней контрольной работы поставленная цель была достигнута. Были получены базовые навыки работы с одномерными массивами, освоены принципы работы со строками как с частным случаем массива.

В процессе работы было столкновение со сложностями, например, во втором задании запрещалось использование стандартных функций для работы со строками, кроме длины строки. Без специальных функций задача казалась невыполнимой, но, изучив интернет-ресурсы, изучив принцип работы со строками как с массивами, задача была решена. Так же было проблематично найти каждое второе вхождение подстроки, но, путем проб и ошибок, был выбран оптимальный вариант решения – счетчик общего количества вхождений, а затем проверка его четности.

Таким образом, преодолев все сложности, а также получив новые знания, была достигнута поставленная цель, выполнены задачи, получены удовлетворяющие результаты.