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-203-52-00

Алферова Софья

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. Формулировка задания (с вариантом)

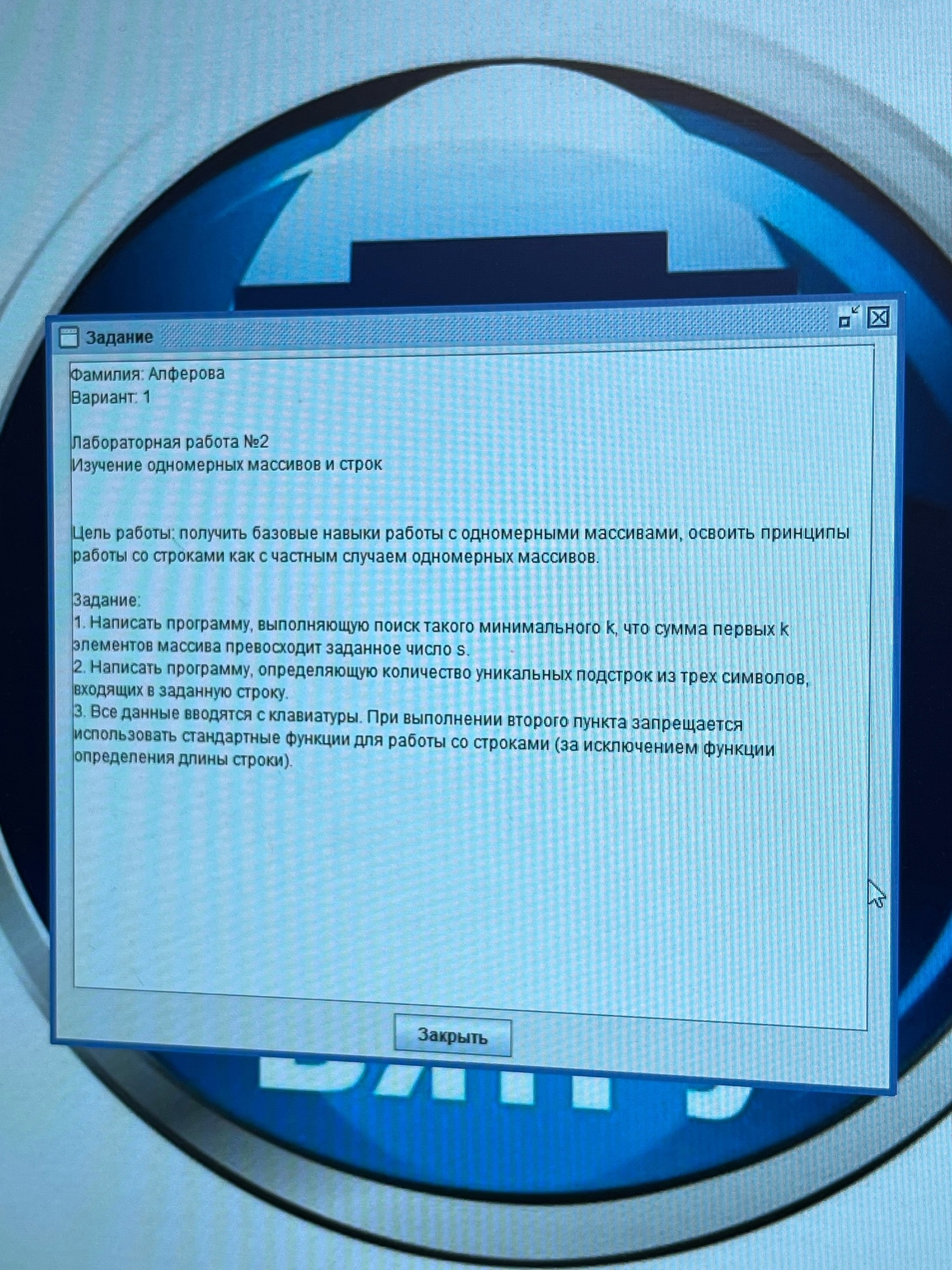


Рис.1

1. Описание алгоритма

Задача 1

* Для начала мы вводим размер массива - N = 10
* Задаем сам массив - arr: **array**[1..N] **of** integer;
* Заполняем массив числами (цикл for с параметром) -

**for** k := 1 **to** N **do**

**begin**

arr[k] := k;

**end**;

* Вводим число s
* Поиск минимального числа k
* Оператор repeat-until в Pascal используется для выполнения блока команд до тех пор, пока указанное условие не станет истинным.

sum := 0;

k := 0;

**repeat**

k := k + 1;

sum := sum + arr[k];

**until** (sum > s) **or** (k = N);

* Вывод результата

Задача 2

* Задаем массив - podstr: **array**[1..1000] **of** string;
* Вводим переменные count, i, j, k: integer;count-счетчик

unique: boolean; //подсчет уникальных значений(логический тип)

* Ввод строки
* Поиск уникальных подстрок
* Цикл for, с использованием оператора break-досрочный выход из цикла, unique- присваем значение true/false

**for** j := i **to** i + 2 **do**

**begin**

a := a + str[j];

**end**;

**for** k := 1 **to** count **do**

**begin**

**if** a = podstr[k] **then**

**begin**

unique := false;

**break**;

**end**;

* Вывод результата

writeln('Количество уникальных подстрок из трех символов: ', count)

1. Схемы алгоритма

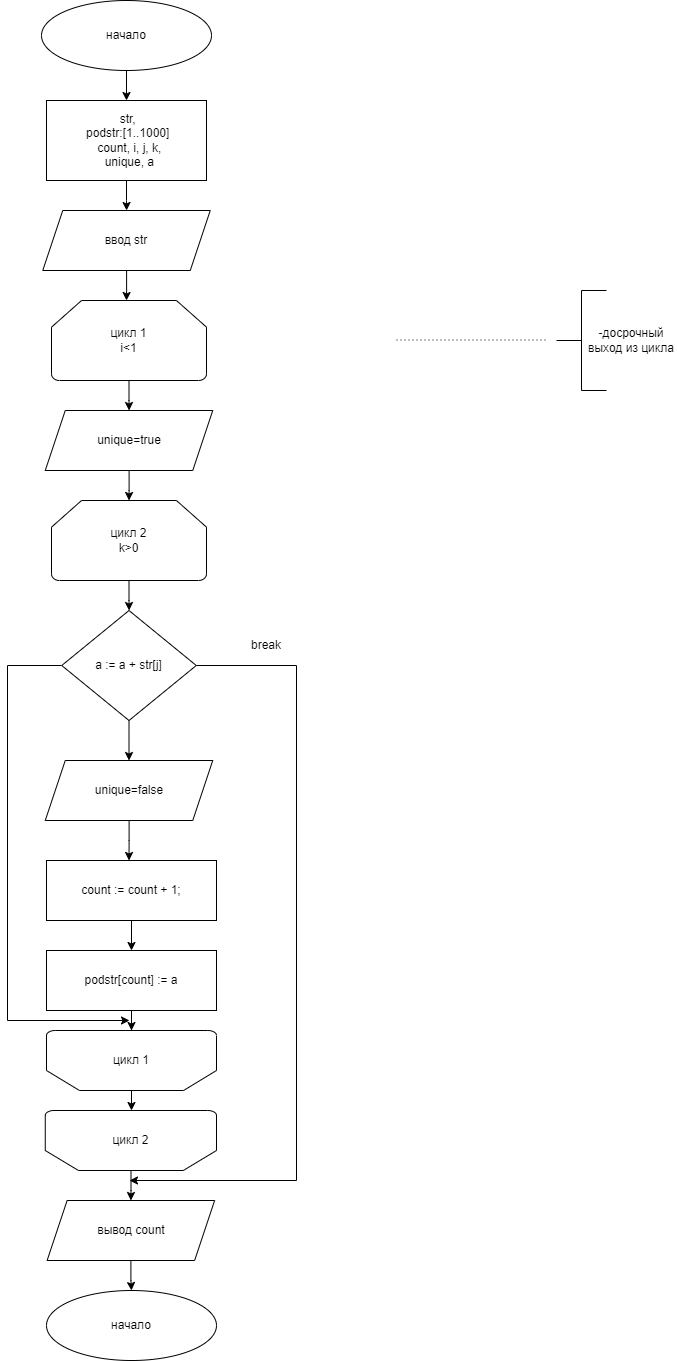
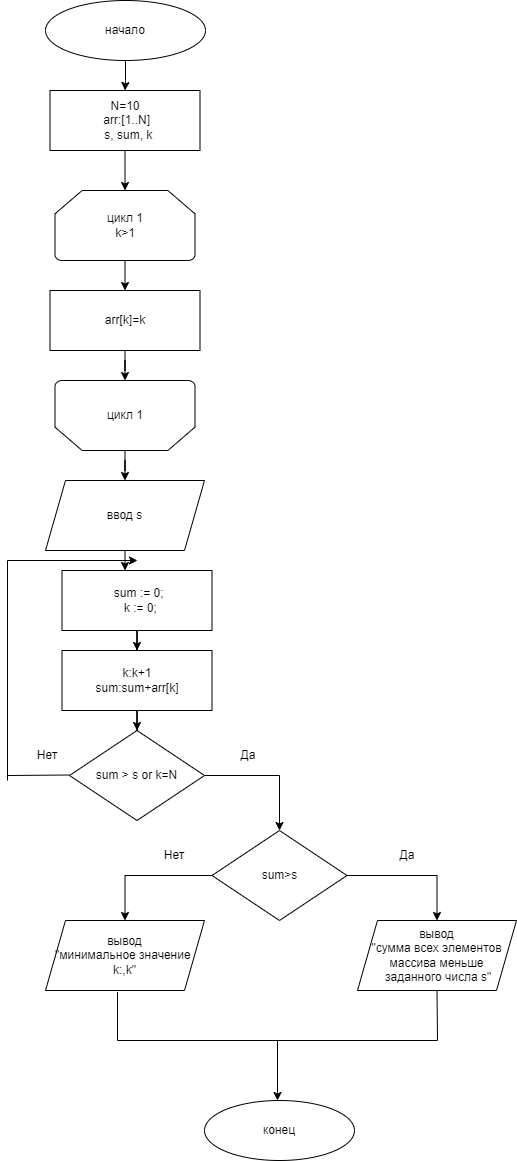


Рис 2. Задача 1 Рис.3 Задача 2

1. Код программы

Задача 1

**program** p1dkr2;

**const**

N = 10;

**var**

arr: **array**[1..N] **of** integer;

s, k, sum: integer;

**begin**

**for** k := 1 **to** N **do**

**begin**

arr[k] := k;

**end**;

write('Введите число s: ');

readln(s);

sum := 0;

k := 0;

**repeat**

k := k + 1;

sum := sum + arr[k];

**until** (sum > s) **or** (k = N);

**if** sum > s **then**

writeln('Минимальное значение k: ',k)

**else**

writeln('Сумма всех элементов массива меньше заданного числа s.');

**end**.

Задача 2

**program** p2dkr2;

**var**

str: string;

podstr: **array**[1..1000] **of** string;

count, i, j, k: integer;

unique: boolean;

a: string;

**begin**

// ввод строки

write('Введите строку: ');

readln(str);

// поиск уникальных подстрок

count := 0;

**for** i := 1 **to** length(str) - 2 **do**

**begin**

unique := true;

a := ”;

**for** j := i **to** i + 2 **do**

**begin**

a := a + str[j];

**end**;

**for** k := 1 **to** count **do**

**begin**

**if** a = podstr[k] **then**

**begin**

unique := false;

**break**;

**end**;

**end**;

**if** unique **then**

**begin**

count := count + 1;

podstr[count] := a;

**end**;

**end**;

// вывод результата

writeln('Количество уникальных подстрок из трех символов: ', count);

**end**.

1. Результат выполнения программы

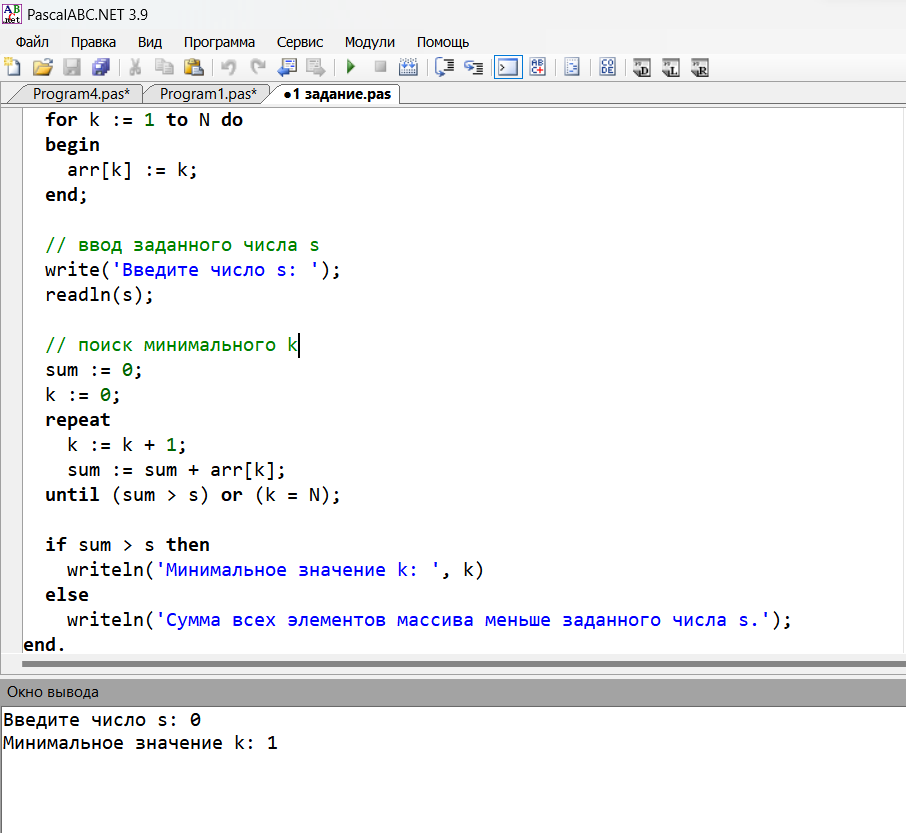


Рис.4 Программа 1

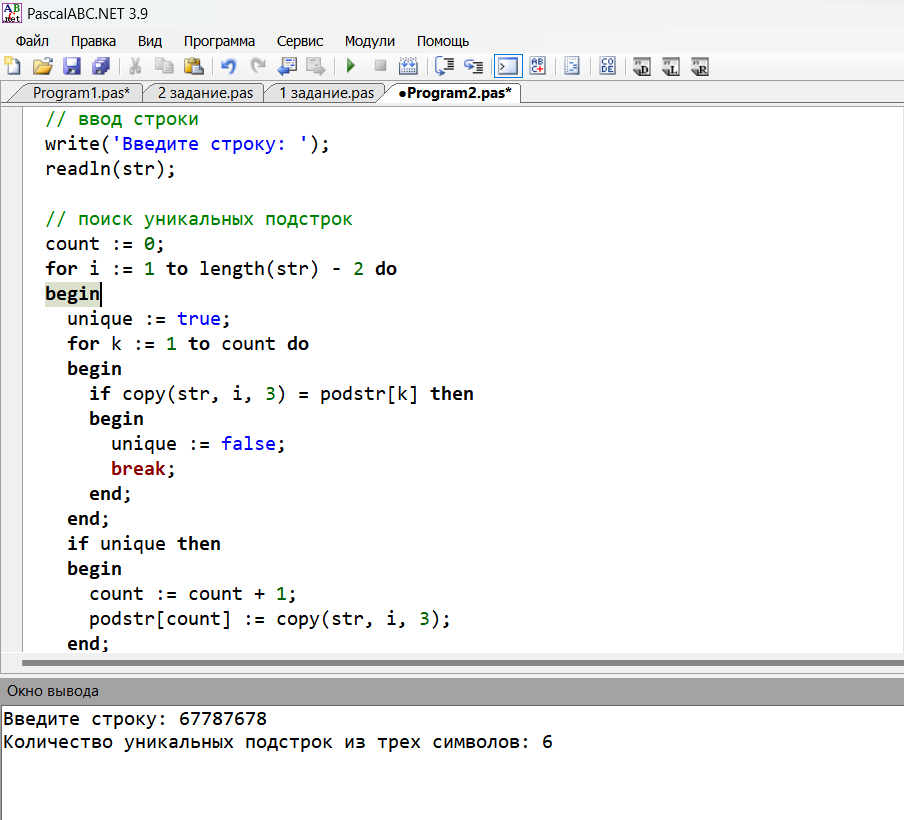


Рис.5 Программа 2

1. Вывод

Вывод: Таким образом, были получены базовые навыки работы с одномерными массивами и освоены принципы работы со строками как с частным случаем одномерных массивов. Было выполнено написание двух программ и построено две схемы алгоритмов. При выполнении домашней контрольной работы возникли трудности: во 2 программе цикл не останавливался и повторял одни и те же действия, соответственно программа работала не правильно. Необходимо было изучить новый оператор break-досрочный выход из цикла.