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**

**«ОДНОМЕРНЫЕ МАССИВЫ И СТРОКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк- 101-51-00

**Мутовкин Михаил Юрьевич**

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

В отчете должны отображаться:

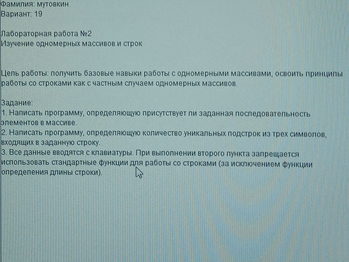
1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Схема алгоритма с комментариями
4. Код программы
5. Результат выполнения программы
6. Вывод
7. Цель: получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.
8. 

Рисунок 1 - Задания

1. Схема алгоритма

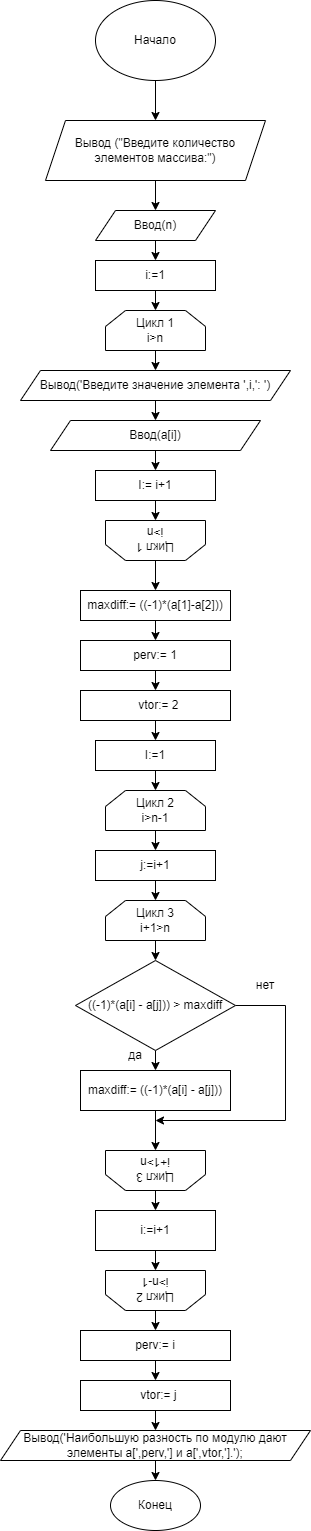


Рисунок 2 – Схема алгоритма к задаче 1

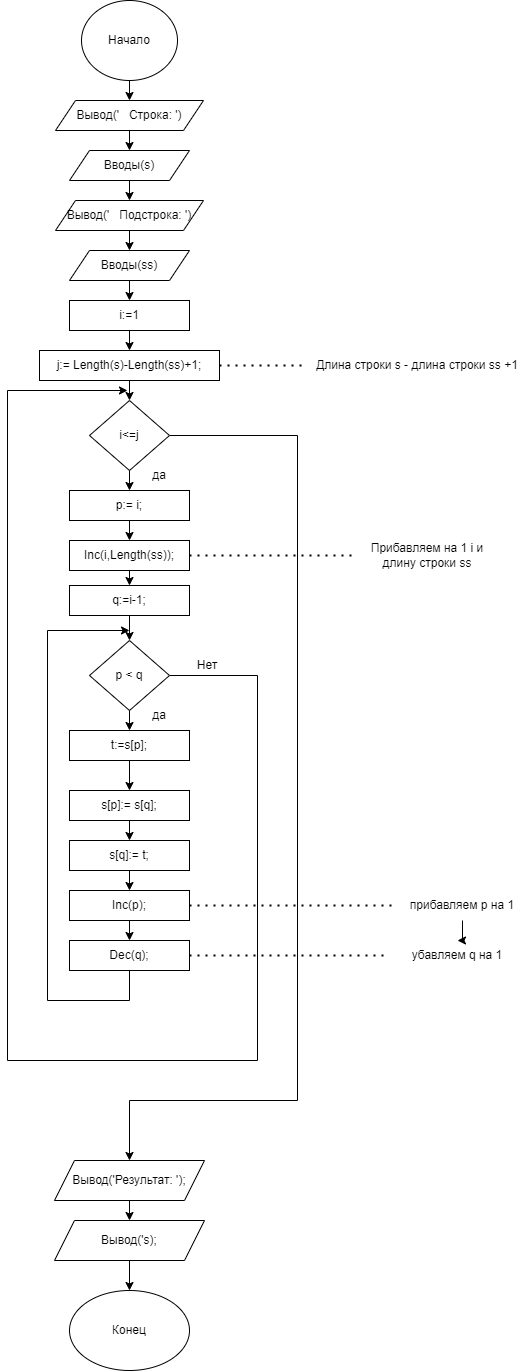


Рисунок 3 - Схема алгоритма к задаче 2

**4.1 begin**

**var** a,b:**array**[1..100] **of** integer;

**var** n,k,j:integer;

randomize;

**repeat**

print('Введите размер массива от 10 до 100 n=');

readln(n);

**until** n **in** [10..100];

print('Массив А');

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

**begin**

a[i]:=random(100);

write(a[i]:4);

**end**;

print;

**repeat**

print('Введите размер последовательности для поиска от 2 до ',n,' k=');

Readln(k);

**until** k **in** [2..n];

writeln('Введите элементы последовательности');

**for var** i:=1 **to** k **do**

readln(b[i]);

**var** i:=1;

**while** (i<n)**and**(a[i]<>b[1]) **do** inc(i);

**if** i=n **then** print('Последовательности в массиве нет')

**else**

**begin**

j:=i;

**while**(j<=k)**and**(b[j]=a[j]) **do** inc(j);

**if** j>k **then** print('Последовательность в массиве есть')

**else** print('Последовательности в массиве нет')

**end**;

**end**.

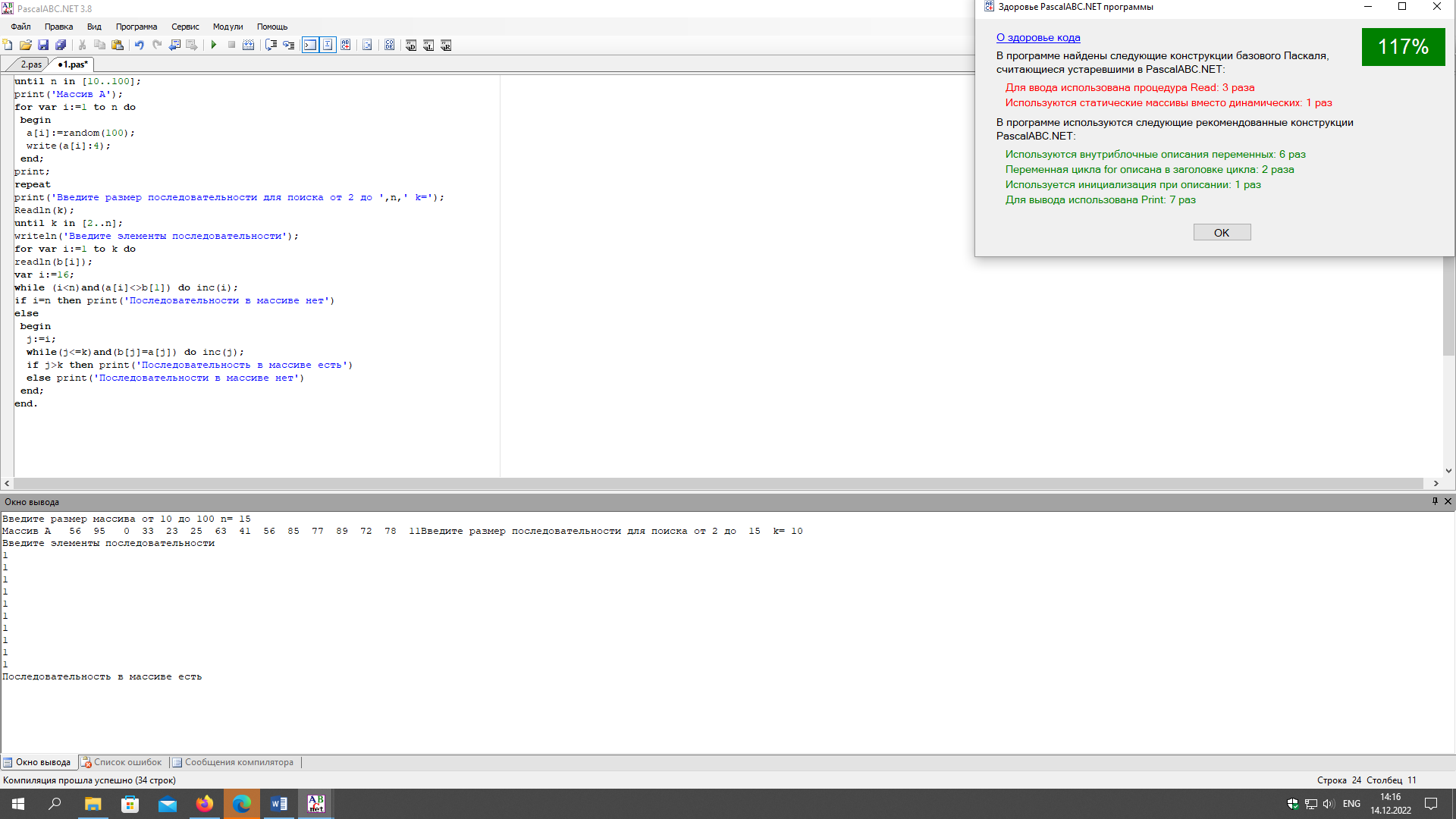


Рисунок 4 - Здоровье кода 1 программы

**4.2 begin**

**var** a :string;

**var** s,p,z:longint;

**var** flag: Boolean;

writeln('Введите строку');

read(a);

z:=0; s:=length(a) ;

**begin**

**for var** k:=0 **to** s-3 **do**

**begin**

p:=0;

flag:=false;

**while** p<s-3 **do begin**

**if** ( k+1=p+1) **and** (k+2=p+2) **and** (k+3=p+3) **then**

p:=p+1;

**if** (a[s-2]=a[k+1]) **and** (a[s-1]=a[k+2])

**and** (a[s]=a[k+3])**and** (k+1<>s-2) **and** (k+2<>s-1) **and** (k+3<>s) **then**

flag:=true;

**if** (a[k+1]=a[p+1]) **and** (a[k+2]=a[p+2]) **and** (a[k+3]=a[p+3]) **then**

flag:=true;

p:=p+1;

**end**;

**if** flag=false **then**

z:=z+1;

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

**end**;

**end**;

**end**.

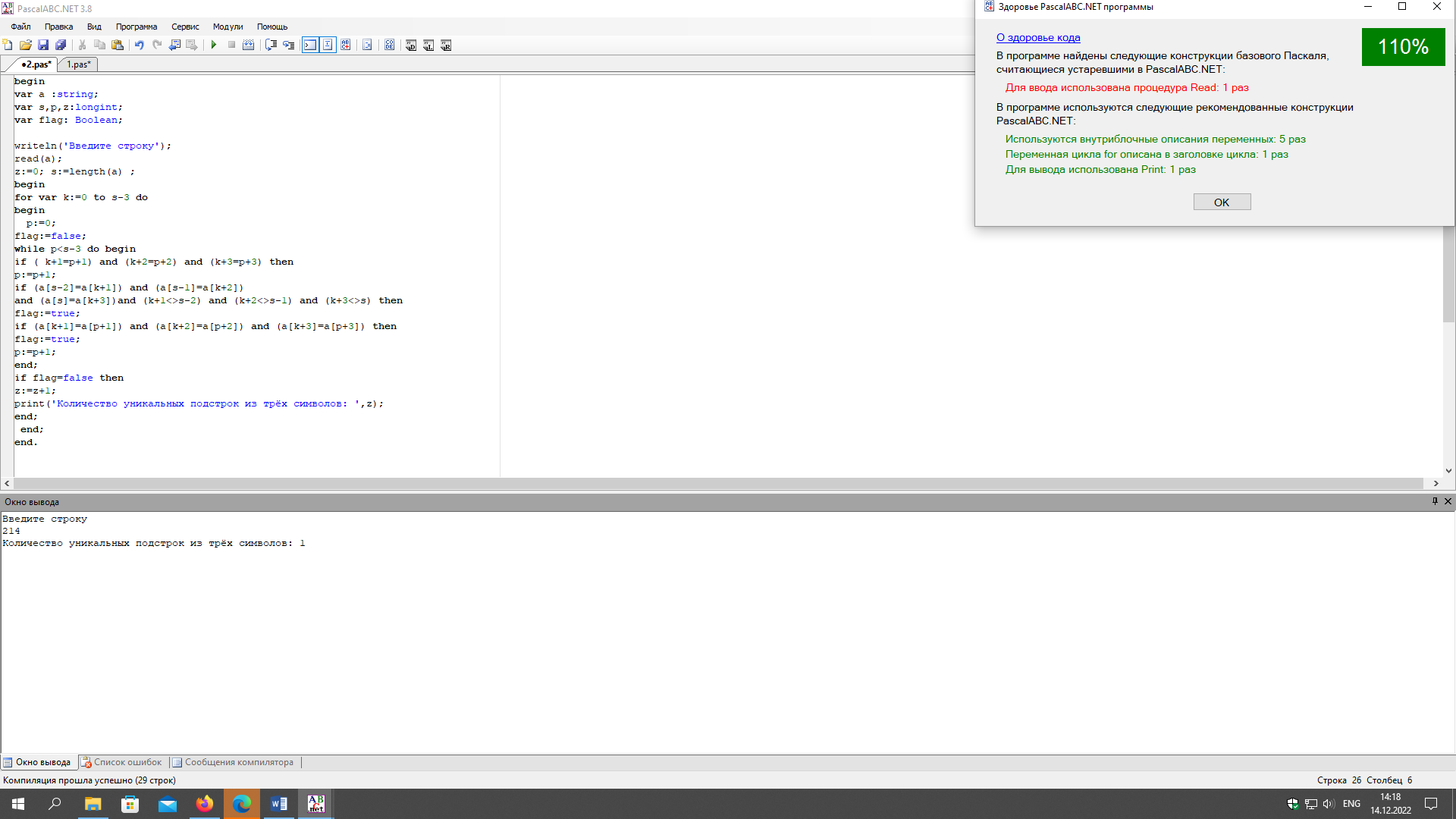


Рисунок 5 - Здоровье кода 2 программы

5.

Рисунок 6 – Значения y из программы 1

Рисунок 7 – Значения y из программы 2

6. В ходе работы была получена программы, исполняющие действия с одномерными массивами.

В ходе работы были получены знания о Здоровье Паскаль, как оптимизировать код на Паскале так, чтобы его уровень здоровья был 100% и выше. Программы написаны без использования встроенных функиций. Задания были оптимизированы под Ввод строки пользователем.

В результате получены программы, первая - выводит два элемента массива с условием, что их разность по модулю максимальна. А вторая - инверсирует подстраку и подставляет ее в строку.