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

Усатова Ульяна Николаевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

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

Вариант: 20

Задание:

1. ﻿﻿﻿Написать программу, выполняющую поиск двух элементов заданного массива, таких, что их разность по модулю минимальна
2. ﻿﻿﻿Написать программу, определяющую количество уникальных подстрок из трех символов, входящих в заданную строку

﻿﻿﻿Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длины строки).

1. **Описание алгоритма**

**№1**

1. Ввод размера массива, максимальный размер 100, размер от 1 до 100.
2. Ввод элементов массива (i:=1 to n-числа, которые будут в массиве) arr.
3. Поиск минимальной разницы между элементами m(минимальная разница)=модуль разницы между двумя элементами массива.
4. Поиск элементов по строке.
5. Вывод минимальных результатов.
6. Поиск пары элементов с минимальной разницей по модулю.
7. Если разность элементов массива равна минимуму, то выводим эту пару массивов.

**№2**

1. Ввод строки a.
2. С помощью цикла for и вложенного в него цикла while вводится подсчет количества уникальных подстрок из 3 символов.
3. Вывод переменной z.
4. **Схема алгоритма с комментариями**

№1

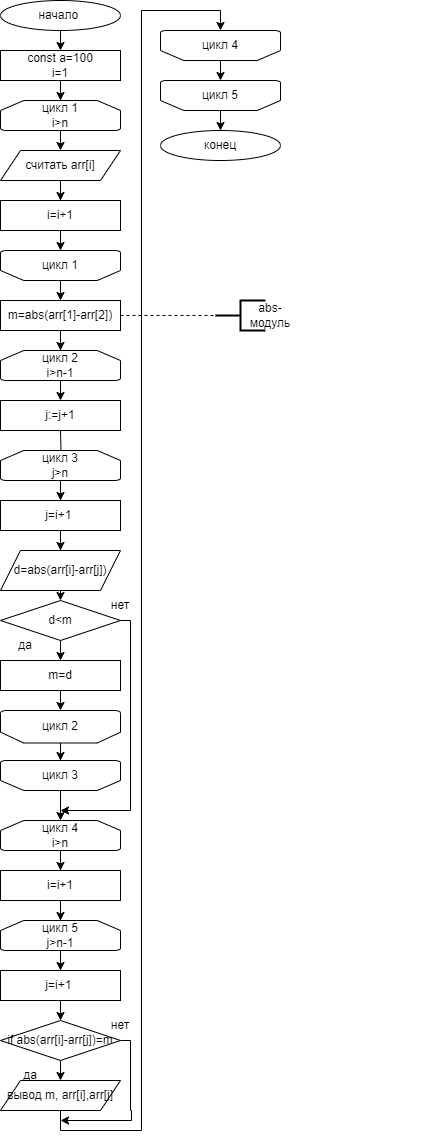


Рис. 1. Схема алгоритма с комментариями.

№2

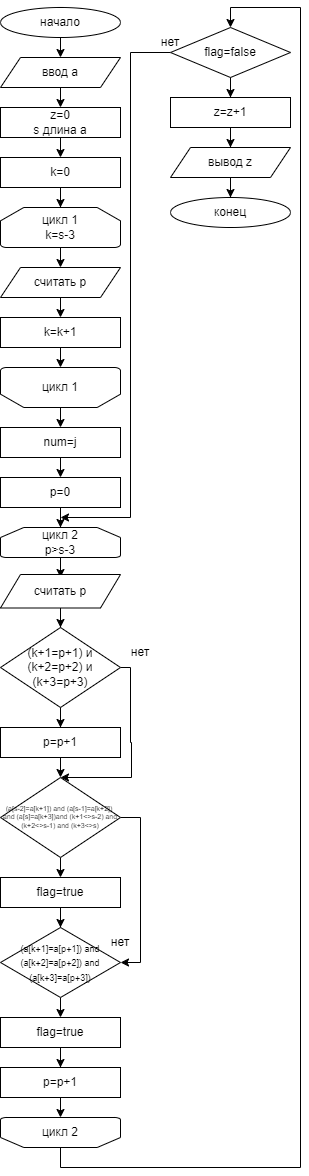


Рис.2. Схема алгоритма с комментариями.

1. **Код программы**

№1

**const**

a= 100;

**var**

arr: **array**[1..a] **of** Integer;

n, i, j, m, d: Integer;

**begin**

write('Введите размер массива: ');

readln(n);

writeln('Введите элементы массива:');

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

**begin**

write('arr[', i, '] = ');

readln(arr[i]);

**end**;

m:= abs(arr[1] - arr[2]);

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

**begin**

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

**begin**

d:= abs(arr[i] - arr[j]);

**if** d< m **then**

m:=d;

**end**;

**end**;

writeln('Минимальная разница элементов: ',m);

writeln('Пары элементов с минимальной разницей по модулю: ');

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

**begin**

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

**begin**

**if** abs(arr[i] - arr[j]) = m **then**

writeln('(', arr[i], ', ', arr[j], ')');

**end**;

**end**;

readln;

**end**.

№2

**var**

a :string;

k,s,p,z:integer;

flag: Boolean;

**begin**

write('Введите строку: ');

read(a);

z:=0;

s:=length(a);

**for** 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;

**end**;

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

**end**.

1. **Результат выполнения программы**

№1

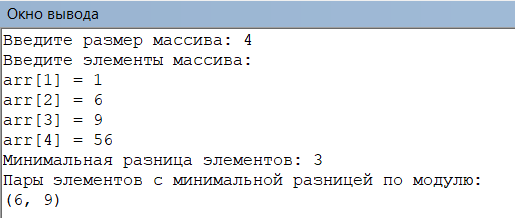


Рис. 3. Результат выполнения программы.

№2

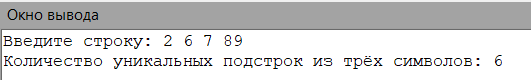


Рис. 4. Результат выполнения программы.

1. **Вывод**

В процессе выполнения данного задания по самостоятельной контрольной работе, мы получили фундаментальные навыки работы с одномерными массивами и освоили принципы работы со строками, рассматривая их как особый случай одномерных массивов. Не только успешно справились с поставленной задачей, но также были увлечены и заинтересованы в процессе изучения языка программирования Pascal. Мы прекрасно уяснили применение массивов, циклов и условий, достигли поставленных целей, что в корне заложило в нас уверенность в программировании и стимулировало нас для последующего развития в этой области.

Для создания схемы алгоритмов мы использовали приложение Draw.io. Draw.io — это бесплатный онлайн-сервис, который помогает создавать схемы алгоритмов, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм, поэтому недавно сервис переименовали в Diagrams.net. Но старое название по-прежнему в ходу. Сервисом пользуются разные IT-специалисты — от бизнес-аналитиков до разработчиков и дизайнеров.