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

Шумилов 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. Формулировка задания:**

**Вариант 24**

Задание состоит из трех частей:

1. Написать программу, выполняющую поиск двух элементов заданного массива, таких, что их разность по модулю равна K  
  
 2. Написать программу, определяющую количество уникальных подстрок из трех символов, входящих в заданную строку.  
  
 3. Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длины строки).

**3.1 Описание алгоритма:**

Randomize используется для инициализации генератора случайных чисел, чтобы при каждом запуске программы получать различные значения.

Пользователю предлагается ввести значение K с клавиатуры.

Массив arr размером N заполняется случайными целыми числами в диапазоне от -100 до 100.

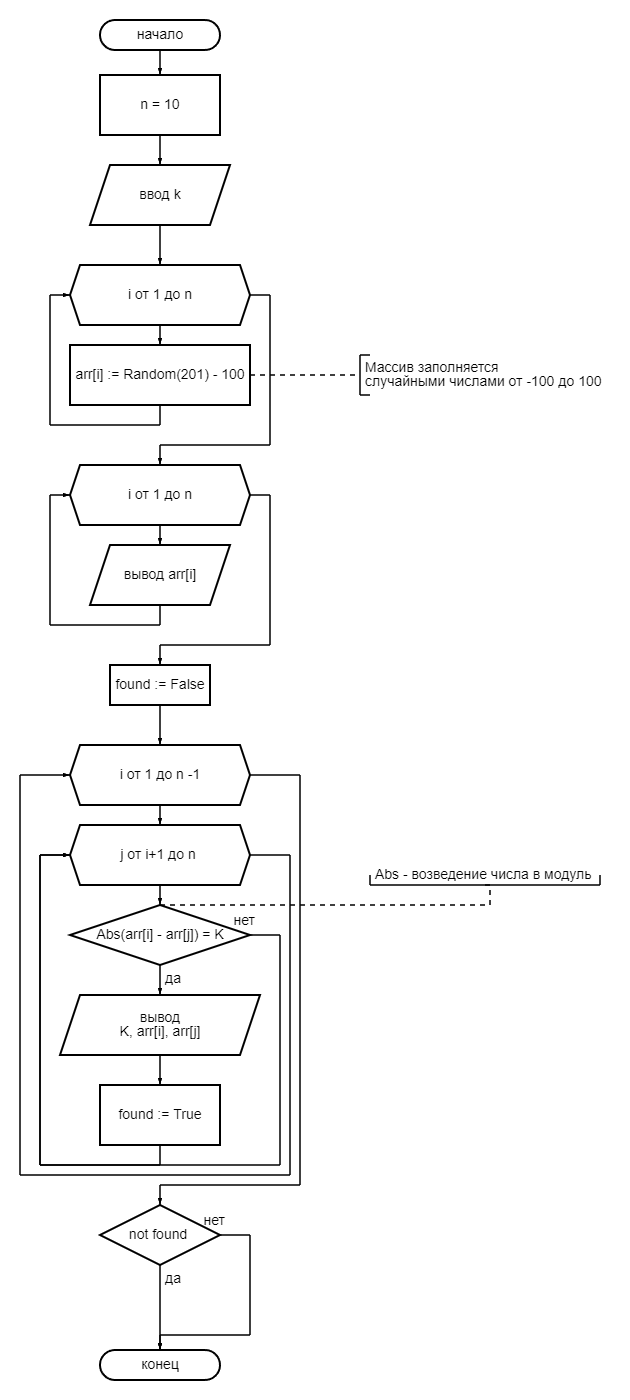
Исходный массив выводится на экран.

Двойной цикл for используется для перебора всех возможных пар элементов массива. Если разность между элементами равна введенному значению K, то эта пара элементов выводится на экран.

Если найдена хотя бы одна пара с разностью K, то программа выводит эти пары на экран. В противном случае выводится сообщение о том, что таких пар нет.

readln используется для ожидания ввода пользователя перед закрытием консоли.

**4.1** **Схема алгоритма с комментариями:**



**5.1 Код программы:**

**program** z\_1;

**const**

N = 10; // Размер массива

**var**

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

i, j, K: Integer;

found: Boolean;

**begin**

write('Введите значение K: ');

readln(K);

write('Ввод элементов массива: ');

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

readln(arr[i]);

writeln('Исходный массив:');

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

write(arr[i], ' ');

writeln;

found := False;

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

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

**if** Abs(arr[i] - arr[j]) = K **then**

**begin**

writeln('Найдены элементы с разностью ', K, ': ', arr[i], ' и ', arr[j]);

found := True;

**end**;

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

writeln('Нет элементов с разностью ', K);

readln;

**end**.

**6.1. Результат выполнения программы**

Введите значение K: 1

Ввод элементов массива:

6

1

2

34

5

6

7

8

9

245

Исходный массив:

6 1 2 34 5 6 7 8 9 245

Найдены элементы с разностью 1: 6 и 5

Найдены элементы с разностью 1: 6 и 7

Найдены элементы с разностью 1: 1 и 2

Найдены элементы с разностью 1: 5 и 6

Найдены элементы с разностью 1: 6 и 7

Найдены элементы с разностью 1: 7 и 8

Найдены элементы с разностью 1: 8 и 9

**3.2. Описание алгоритма:**  
Ввод строки:

Пользователю предлагается ввести строку с клавиатуры, которая сохраняется в переменной org.

Проверка длины строки:

Проверяется длина введенной строки. Если длина меньше 3 или больше значения константы n (в данном случае 1000), программа выводит сообщение о том, что длина строки слишком большая или маленькая, и завершает выполнение.

Преобразование строки в массив символов:

Строка org преобразуется в массив символов arr, где каждый символ строки занимает соответствующий элемент массива.

Поиск уникальных подстрок:

Используется два вложенных цикла for, чтобы перебрать все возможные комбинации подстрок из трех символов.

Используется массив b, чтобы отмечать подстроки, которые уже были обработаны.

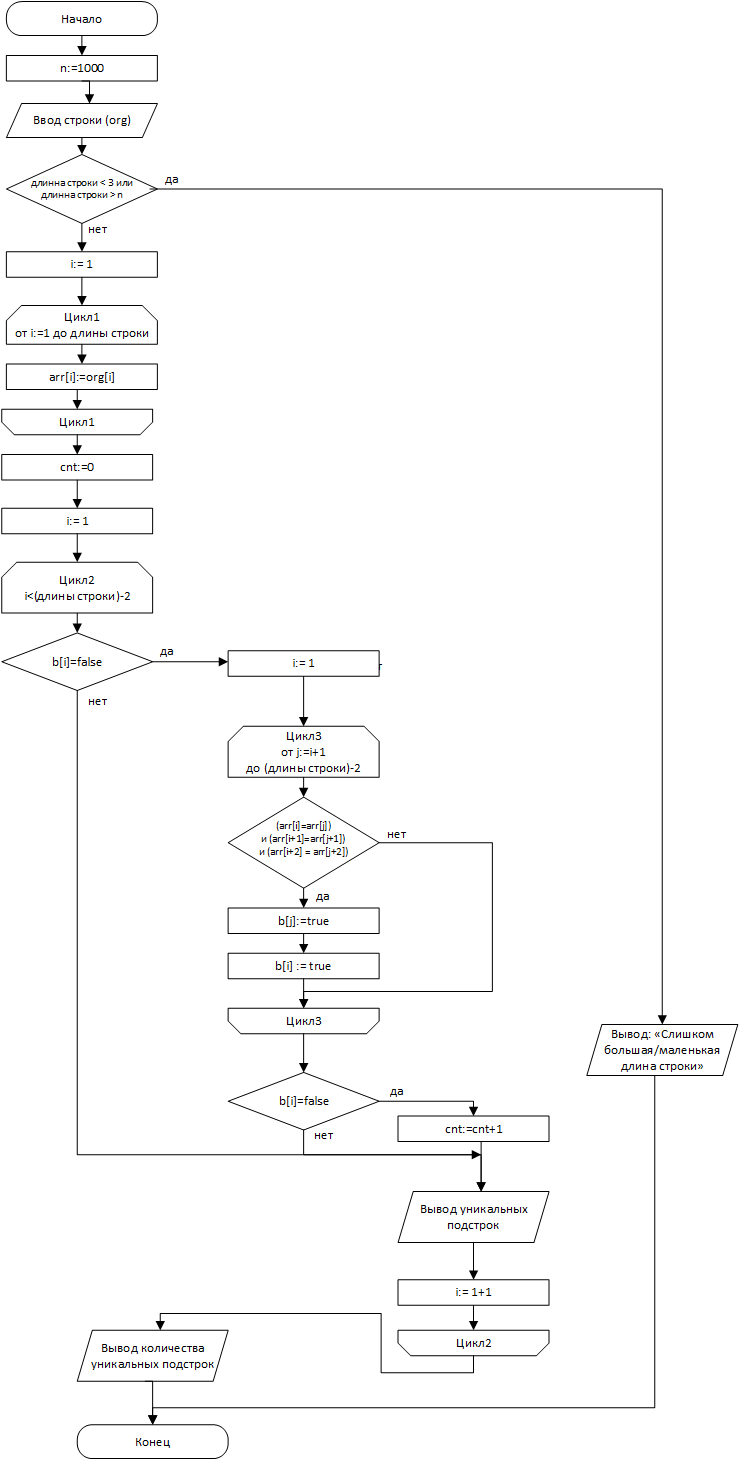
Если подстроки совпадают (по трем последовательным символам), то оба индекса отмечаются в массиве b.

Если подстрока не была отмечена, она считается уникальной, и индекс i используется для вывода этой подстроки.

Подсчитывается количество уникальных подстрок.

Вывод результата:

Выводится количество уникальных подстрок и сами уникальные подстроки.

**4.2. Схема алгоритма с комментариями:** 

**5.2. Код программы:** **program**

**program** z\_2;

**const**

n = 1000;

**var**

org: string;

arr: **array**[1..n] **of** char;

b: **array**[1..n] **of** boolean;

i, j, cnt: integer;

**begin**

writeln('Введите строку:');

readln(org);

**if** (length(org) < 3) **or** (length(org) > n) **then begin**

writeln('Слишком большая/маленькая длина строки');

**exit**;

**end**;

**for** i := 1 **to** length(org) **do** arr[i] := org[i];

cnt := 0;

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

**begin**

**if not** b[i] **then begin**

**for** j := i + 1 **to** length(org) - 2 **do**

**begin**

**if** (arr[i] = arr[j]) **and** (arr[i + 1] = arr[j + 1]) **and** (arr[i + 2] = arr[j + 2]) **then begin**

b[j] := true;

b[i] := true;

**end**;

**end**;

**if not** b[i] **then begin**

cnt := cnt + 1;

writeln('Уникальная подстрока: ', arr[i], arr[i + 1], arr[i + 2]);

**end**;

**end**;

**end**;

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

**end**.

**6.2. Результат выполнения программы**

Введите строку:

Я очень сильно люблю ОАИП

Уникальная подстрока: Я о

Уникальная подстрока: оч

Уникальная подстрока: оче

Уникальная подстрока: чен

Уникальная подстрока: ень

Уникальная подстрока: нь

Уникальная подстрока: ь с

Уникальная подстрока: си

Уникальная подстрока: сил

Уникальная подстрока: иль

Уникальная подстрока: льн

Уникальная подстрока: ьно

Уникальная подстрока: но

Уникальная подстрока: о л

Уникальная подстрока: лю

Уникальная подстрока: люб

Уникальная подстрока: юбл

Уникальная подстрока: блю

Уникальная подстрока: лю

Уникальная подстрока: ю О

Уникальная подстрока: ОА

Уникальная подстрока: ОАИ

Уникальная подстрока: АИП

Количество уникальных подстрок из трех символов: 23

**7.**  **Вывод:**

В ходе данной работы были освоены основы манипулирования одномерными массивами и строками на примере языка программирования Pascal. Разработанная программа подчеркивает умение эффективно работать с массивами символов для обработки строк и использовать индексацию для доступа к конкретным элементам.

Особый акцент сделан на использовании логических массивов, что позволяет эффективно управлять обработанными данными и избегать повторных проверок. Полученные навыки являются универсальными и применимы в различных областях программирования, особенно в обработке и анализе текстовых данных.

Этот опыт не только обогащает знания о языке Pascal, но также является переносимым на другие языки программирования, расширяя умения по структурированию и обработке данных. Приобретенные навыки работы с одномерными массивами и строками представляют собой важный этап в процессе освоения основ программирования.