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

Обухов Илья Николаевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы:**

Цель данной работы заключается в получении базовых навыков работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.

1. **Формулировка задания:**

Вариант: 17

1. Написать программу, выполняющую поиск k-ой статистики в массиве.

2. Написать программу, определяющую кол-во уникальных подстрок из трёх 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).

1. **Описание алгоритма:**
2. Выполнение программы:

1) Устанавливаем размер массива **r**, который составляет 10 элементов.

2) Определяем переменные: массив **arr** для хранения введенных значений, **k** для указания k-ой статистики, и **i**, **j**, **x** для управления циклами и временным хранением значений.

3) Вводятся значения массива с клавиатуры, и эти значения сохраняются в массиве **arr**.

4) Вводится значение k - порядковой статистики, которую программа будет находить.

5) Проверка, находится ли **k** в допустимом диапазоне, если нет, программа выводит сообщение об ошибке и завершает выполнение.

6) Выполняется сортировка для массива **arr**, ищется k-ая статистика.

7) Отсортированный массив выводится на экран.

8) Вывод k-ой статистики. k-ой ста\\

тистики выводится на экран.

Начало формы

1. Выполнение программы:

1) Пользователь вводит строку и ее длина строки сохраняется в переменной dl.

2) Затем начинается цикл, в котором переменная i изменяется от 0 до dl-3.

3) Внутри этого цикла выполняется еще один цикл, в котором переменная j изменяется от 0 до dl-3.

4) Внутри второго цикла проверяется, равны ли i+1, i+2 и i+3 j+1, j+2 и j+3 соответственно. Если равны, то j увеличивается на 1.

5) Затем проверяется, равны ли последние три символа строки s с символами s[i+1], s[i+2] и s[i+3], и не являются ли они последними тремя символами строки. Если условие выполняется, переменная n устанавливается в true.

6) Затем проверяется, равны ли символы s[i+1], s[i+2] и s[i+3] символам s[j+1], s[j+2] и s[j+3]. Если условие выполняется, переменная n устанавливается в true.

7) После окончания второго цикла, если n остается false, то переменная z увеличивается на 1.

8) По завершении всех циклов выводится количество уникальных подстрок из трех символов.

1. **Схемы алгоритмов:**

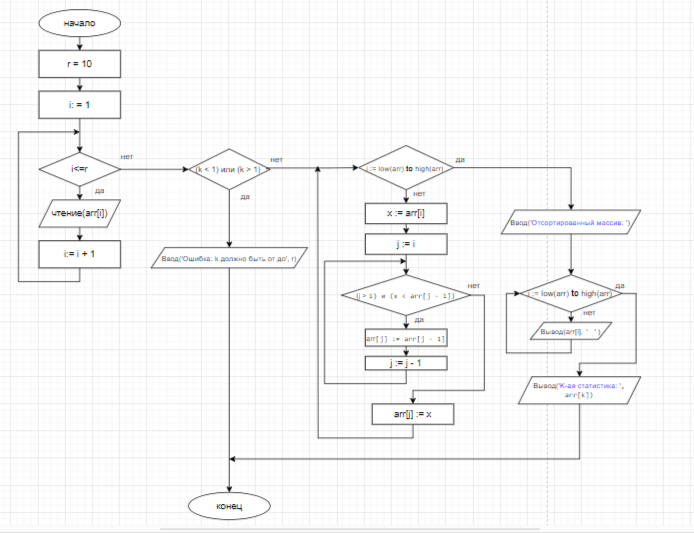


Рисунок 1 – Схема задания.

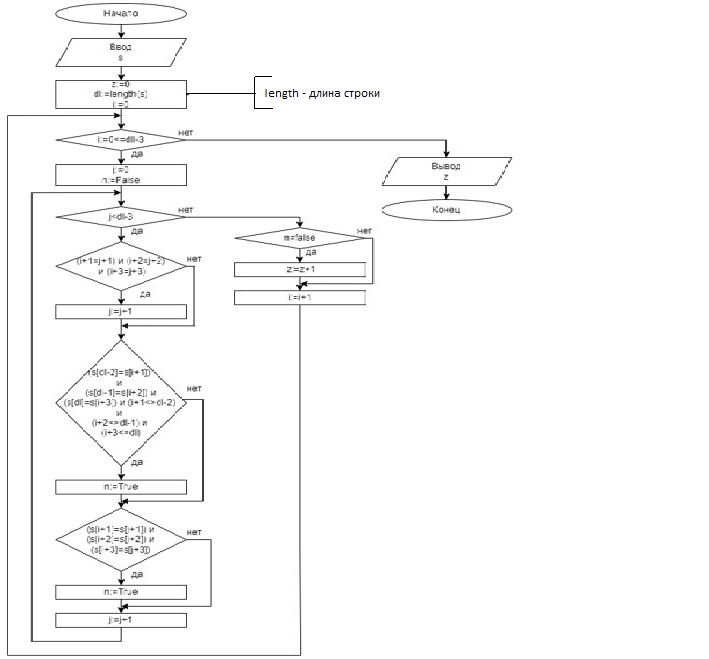


Рисунок 2 – Схема задания.

1. **Код программы:**
2. Поиск k-ой статистики в массиве:

**program** dkr2z1;

**const**

r = 10; // размер массива

**var**

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

k: integer; // к-ая статистика

i, j, x: integer;

**begin**

writeln('Введите массив: '); // ввод массива с клавиатуры

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

read(arr[i]);

write('Введите k: '); // ввод k

readln(k);

**if** (k < 1) **or** (k > r) **then** // проверка корректности введенного k

**begin**

writeln('Ошибка: k должно быть от 1 до ', r);

**exit**;

**end**;

**for** i := low(arr) **to** high(arr) **do** // алгоритм поиска k-ой статистики

**begin**

x := arr[i];

j := i;

**while** (j > 1) **and** (x < arr[j - 1]) **do**

**begin**

arr[j] := arr[j - 1];

j := j - 1; // уменьшает j на единицу

**end**;

arr[j] := x;

**end**;

write('Отсортированный массив: '); // вывод отсортированного массива

**for** i := low(arr) **to** high(arr) **do**

write(arr[i], ' ');

writeln;

writeln('K-ая статистика: ', arr[k]); // вывод k-ой статистики

**end**.

1. Определение кол-ва уникальных подстрок из трёх символов, входящих в заданную строку:

**program** dkr2z2;

**var**

s: string;

i, dl, j, z: integer;

n: Boolean;

**begin**

writeln('Введите строку: ');

readln(s);

z:=0;

dl:=length(s);

**for** i:=0 **to** dl-3 **do**

**begin**

j:=0;

n:=false;

**while** j<dl-3 **do**

**begin**

**if** (i+1=j+1) **and** (i+2=j+2) **and** (i+3=j+3) **then**

j:=j+1;

**if** (s[dl-2]=s[i+1]) **and** (s[dl-1]=s[i+2])

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

n:=true;

**if** (s[i+1]=s[j+1]) **and** (s[i+2]=s[j+2]) **and** (s[i+3]=s[j+3]) **then**

n:=true;

j:=j+1;

**end**;

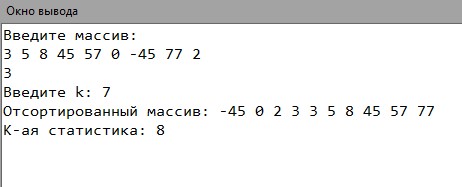
**if** n=false **then**

z:=z+1;

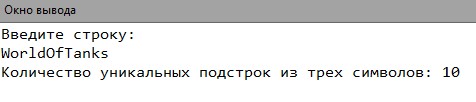
**end**;

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

1. **Результаты выполнения программ:**
2. Результат выполнения 1-ой программы:



1. Результат выполнения 2-ой программы:



1. **Вывод:**

По моему мнению, работа прошла успешно. В процессе выполнения данного задания по домашней 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. Сервисом пользуются разные IT-специалисты — от бизнес-аналитиков до разработчиков и дизайнеров.