**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

Долгих Дмитрий Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

Получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.

**Формулировка задания**

Вариант 9

1. Написать программу, выполняющую поиск среднеарифметического в заданном массиве.
2. Написать программу, удаляющую из строки первое и последнее вхождение заданной подстроки.
3. Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длины строки).

**Описание алгоритма**

Программа №1

В программе был использован цикл *for* для ввода чисел в массив и суммирование всех чисел в массиве, в конце высчитывалось среднеарифметическое в массиве

Программа №2

В программе были использованы условные операторы *if…else* и циклы с предусловием *while*, с помощью которых находятся и удаляются первое и последнее вхождения подстрок из строки.

Сначала запускается цикл, который проходит по каждому символу строки ‘s’. Внутри цикла запускается еще один цикл, который сравнивает символы строки ‘s’ с символами подстроки ‘ps’. Если найдено совпадение, то индекс ‘k’ увеличивается на 1. Если ‘k’ становится равным длине подстроки ‘ps’, то значит, найдено вхождение подстроки, и обновляются переменные ‘iFirst’ и ‘iLast’. Происходит смещение индекса ‘i’ на количество символов подстроки ‘ps’. Если совпадение не найдено, индекс ‘i’ увеличивается.

Если найдено хотя бы одно вхождение подстроки, то создается строка ‘t’. Запускается цикл, который проходится по каждому символу строки ‘s’. Если текущий индекс соответствует началу или концу вхождения подстроки, то индекс ‘i’ увеличивается на длину подстроки ‘ps’. В противном случае, текущий символ добавляется к строке ‘t’. Исходная строка ‘s’ заменяется на строку ‘t’.

**Схема алгоритма с комментариями**

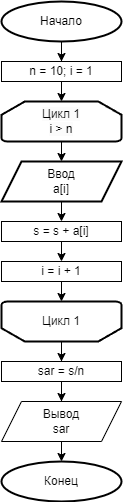


Рис. 1 «Схема алгоритма программы №1»

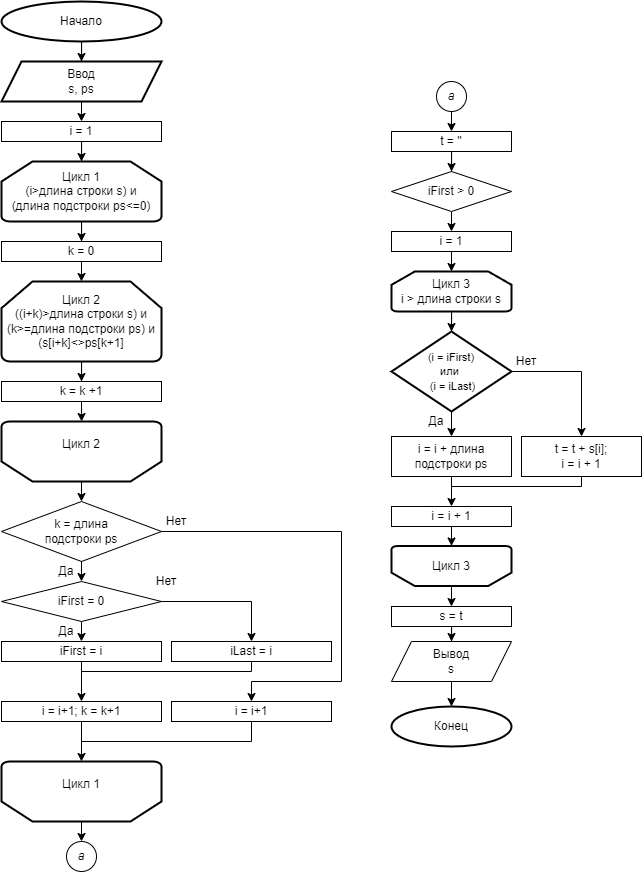


Рис. 2 «Схема алгоритма программы №2»

**Код программы**

Программа №1

**program** dkr2\_1;

**const**

n = 10;

**var**

a: **array** [1..n] **of** integer;

i, s, sar: integer;

**begin**

writeln('Введите массив:');

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

**begin**

read(a[i]);

s+= a[i];

**end**;

writeln('Среднеарифметическое в массиве: ', sar:= s/n)

**end**.

Программа №2

**program** dkr2\_2;

**var**

s, ps, t: string;

i, iFirst, iLast, k: integer;

**begin**

write('Введите строку: ');

readln(s);

write('Введите подстроку: ');

readln(ps);

// нахождение первого и последнего вхождения //

i:= 1;

**while** (i<=length(s)) **and** (length(ps)>0) **do**

**begin**

k:= 0;

**while** ((i + k)<=length(s)) **and** (k<length(ps)) **and** (s[i+k] = ps[k+1]) **do** inc(k);

**if** k=Length(ps) **then**

**begin**

**if** iFirst=0 **then** iFirst:= i

**else** iLast:= i;

inc(i, k);

**end**

**else** inc(i);

**end**;

// удаление подстрок из строки //

t:= '';

**if** iFirst>0 **then**

**begin**

i:= 1;

**while** i<=Length(s) **do**

**begin**

**if** (i=iFirst) **or** (i=iLast) **then** inc(i, Length(ps))

**else**

**begin**

t:= t+s[i];

inc(i);

**end**;

**end**;

s:= t;

**end**;

writeln(s);

**end**.

**Результат выполнения программы**

Программа №1

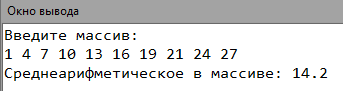


Рис. 3 «Результаты выполнения программы №1»

Программа №2

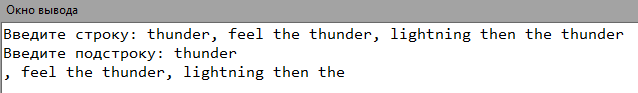


Рис. 4 «Результаты выполнения программы №2»

**Вывод**

Данная работа прошла успешно, в ней мы получили базовые навыки работы с одномерными массивами, освоили принципы работы со строками как с частным случаем одномерных массивов. В работе использовались циклы *for* и *while,* а также условный оператор *if…else*. Цикл *for* применялся для ввода элементов массива, а цикл *while* использовался для поиска и удаления подстроки в строке. Условный оператор *if...else* использовался для определения первого и последнего вхождения подстроки, а также для управления процессом удаления подстроки из строки.

Во время выполнения работы возникли трудности с написанием кода и построением схемы алгоритма. Некоторые из таких проблем – нахождение первого и последнего вхождения подстроки в строке и их удаление из строки. В ходе изучения этой проблемы и поиска решения в Интернете она была решена, пусть и не самым удобным и эффективным способом – удаление вхождений через добавление новой строки, в которую записывалась та же строка, но без первого и последнего вхождения подстроки. Скорее всего, данную задачу можно было решить другим, более удобным способом, но ограничение в виде запрета на использование стандартных функций для работы со строками сильно затрудняло решение задачи.

В итоге, работа улучшила знания в языке программирования Pascal, научила работать с одномерными массивами и строками как частным случаем одномерного массива и также пользоваться циклами *for* и *while*.