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-101-51-00

Долинин Владислав Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

1. **Цель работы:** получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.
2. **Вариант 2.**

Формулировка заданий:

1. Написать программу, определяющую является ли заданный массив монотонным.
2. Написать программу, определяющую является ли заданная строка подпоследовательностью исходной строки.
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. Описание алгоритма

Программа номер 1

* Пользователь вводит размер массива n.
* Если n меньше или равно 1, массив считается монотонным
* Программа создает массив a размером n.
* Пользователь вводит n целых чисел, которые сохраняются в массиве.
* Программа предполагает, что массив монотонный.
* Сначала определяем, как направлен массив (возрастает или убывает) с помощью разности первых двух элементов.
* Затем проверяем все остальные элементы:
* Если направление меняется, программа устанавливает ismon в False и прекращает проверку.
* Если ismon остался равным True, выводим сообщение, что массив монотонный.
* Если ismon стал False, выводим сообщение, что массив не монотонный.

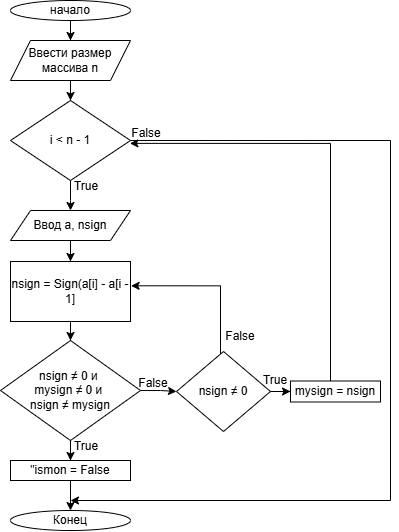
Программа номер 2

* Пользователь вводит две строки: s (основная) и s1 (подстрока).
* Используется функция Pos(s1, s), чтобы проверить, есть ли s1 в s.
* Если Pos возвращает больше 0, выводится сообщение, что s1 является подстрокой s.
* Если 0 - выводится сообщение, что s1 не является подстрокой.

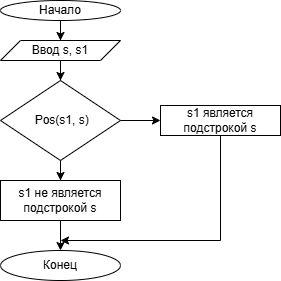
Программа номер 3

* sourceStr — исходная строка, в которой проверяется наличие подпоследовательности.
* subStr — строка, которую нужно проверить на подпоследовательность.
* Определяются длины строк sourceLen и subLen с помощью функции Length.
* i — индекс для прохода по символам sourceStr, начинается с 1.
* j — индекс для прохода по символам subStr, также начинается с 1.
* Цикл работает, пока оба условия выполняются.
* Если текущие символы sourceStr[i] и subStr[j] совпадают, индекс j увеличивается на 1.
* В любом случае индекс i увеличивается на 1, переходя к следующему символу исходной строки.
* Если j > subLen, это значит, что все символы строки subStr найдены в строке sourceStr в указанном порядке. Строка subStr является подпоследовательностью.
* Если цикл завершается, а j <= subLen, значит, не все символы строки subStr были найдены в sourceStr. Подпоследовательности нет.
* Если isSubsequence = true, выводится сообщение: "Строка является подпоследовательностью."

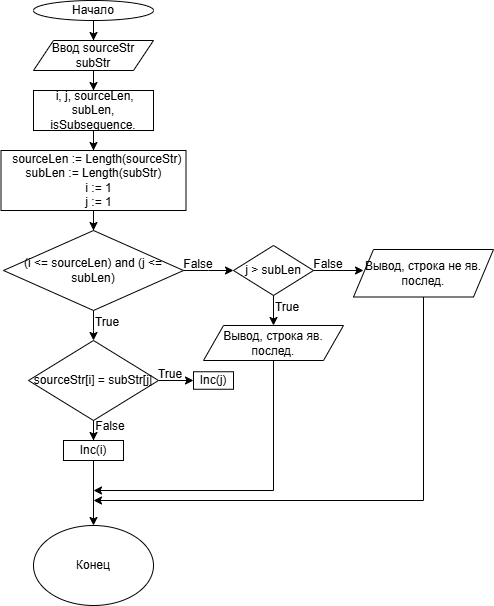
1. **Схема алгоритма.**Программа №1

****

Программа №2



Программа №3



# 5. Программа №1

*var*

*i, n, mysign, nsign: Integer; ismon: Boolean; a: array of Integer;*

*begin*

*Write('Введите размер массива: ');*

*ReadLn(n);*

*if n <= 1 then begin*

*WriteLn('Массив автоматически считается монотонным.');*

*Exit; end;*

*SetLength(a, n);*

*WriteLn('Введите элементы массива: ');*

*for i := 0 to n - 1 do*

*ReadLn(a[i]);*

*ismon := True; mysign := Sign(a[1] - a[0]);*

*for i := 2 to n - 1 do begin*

*nsign := Sign(a[i] - a[i - 1]);*

*if (nsign <> 0) and (mysign <> 0) and (nsign <> mysign) then begin ismon := False; Break; end;*

*if nsign <> 0 then mysign := nsign;*

*end;*

*if ismon then*

*WriteLn('Массив монотонный')*

*else*

*WriteLn('Массив не монотонный'); end.*

Программа №2

*var s, s1: string; begin*

*Write('Введите 1-ю строку '); ReadLn(s); Write('Введите 2-ю строку: ');*

*ReadLn(s1);*

*if Pos(s1, s) > 0 then*

*WriteLn('Вторая строка является подстрокой первой') else*

*WriteLn('Вторая строка не является подстрокой первой'); End.*

Программа №3

*var*

*sourceStr, subStr: string;*

*i, j, sourceLen, subLen: Integer;*

*isSubsequence: Boolean;*

*begin*

*Write('Введите исходную строку: ');*

*ReadLn(sourceStr);*

*Write('Введите строку для проверки подпоследовательности: ');*

*ReadLn(subStr);*

*sourceLen := Length(sourceStr);*

*subLen := Length(subStr);*

*i := 1;*

*j := 1;*

*while (i <= sourceLen) and (j <= subLen) do*

*begin*

*if sourceStr[i] = subStr[j] then*

*Inc(j);*

*Inc(i);*

*end;*

*isSubsequence := j > subLen;*

*if isSubsequence then*

*WriteLn('Строка является подпоследовательностью.')*

*else*

*WriteLn('Строка не является подпоследовательностью.');*

*end.*

# 6. Результат выполнения программы

Программа номер 1:

Введите размер массива: 1 5 6

Массив автоматически считается монотонным.

Программа номер 2:

Введите 1-ю строку 5 4 3

Введите 2-ю строку: 4 3 5

Вторая строка не является подстрокой первой

Программа номер 3:

Введите исходную строку: 12345

Введите строку для проверки подпоследовательности: 123456789

Строка не является подпоследовательностью.

# 7. Вывод

В данной ДКР были написаны задачи и коды. Сделано описание алгоритма, вывод того, что выдавали программы. Сделана схема алгоритма. Написан вывод