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**

**«Вычисление значения функции»**

**ПО ДИСЦИПЛИНЕ «Основы Алгоритмизации и Программирования»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Макаров Никита Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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:**

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

1. **Формулировка задания:** Вариант 9

1. Написать программу, определяющую является ли подпоследовательность элементов массива, расположенных на чётных позициях монотонной

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).

1. **Описание алгоритма:**

**Задание 1**:

Ввод с клавиатуры массива , задание 2 переменных значения true,проверка чисел в массиве на четность ,далее идет проверка элементов массива на четных позициях ,являются ли все числа монотонными, вывод является ли значение монотонным.

**Задание 2:**

Вводится строка, ввод подстроки пользователем, далее проверяется 1 вхождения подстроки, заданной пользователем, после этого происходит проверка последнего вхождения подстроки, происходит проверка 1 и последнего вхождения а после этого удаление этих подстрок.

1. **Схема алгоритма с комментариями:**

Рисунок 1 для задачи №1

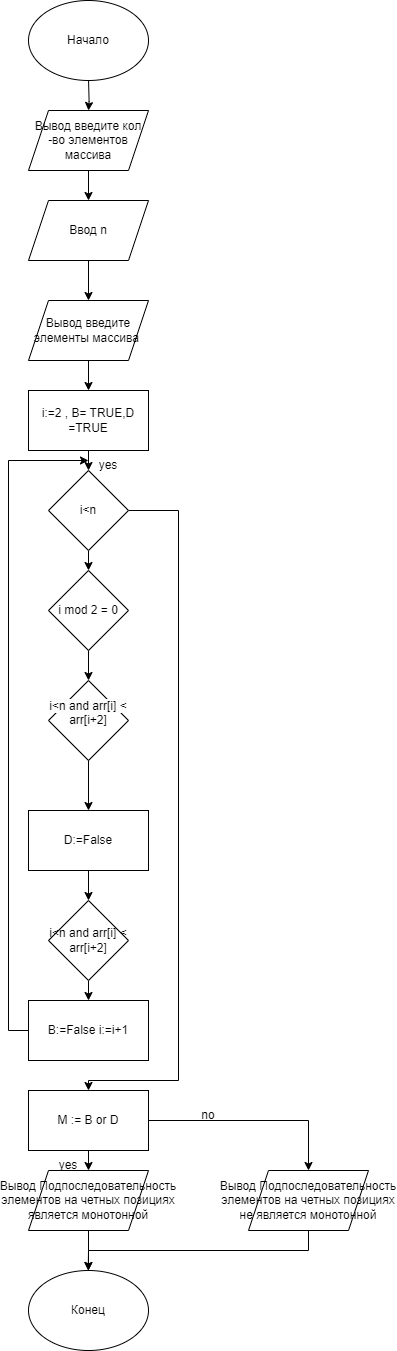


Рисунок 1 – Алгоритм решения задачи №1

Рисунок 2.1 для задачи №2

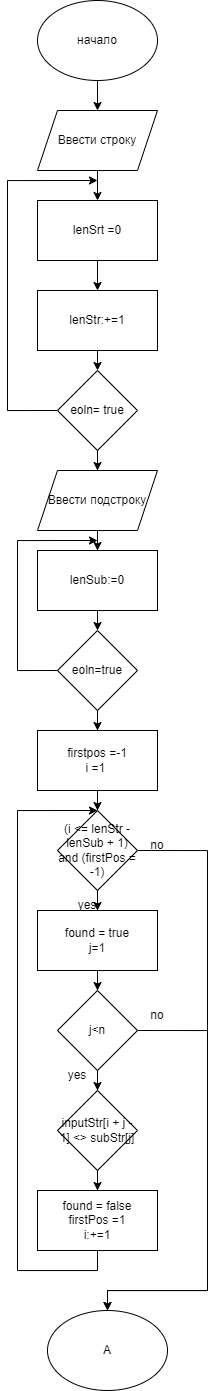


Рисунок 2.1 – Алгоритм решения задачи №2

Рисунок 2.2 для задачи №2

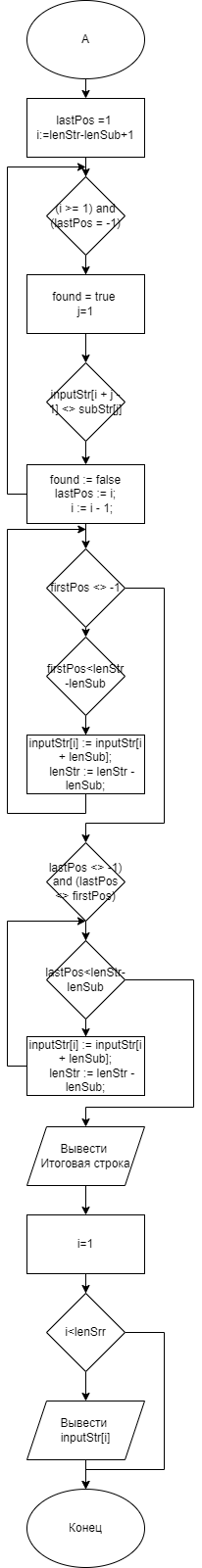


Рисунок 2.2 – Алгоритм решения задачи №2

1. **Код программы**

1 Задание

**var**

arr: **array**[1..20] **of** integer;

n, i: integer;

M: boolean;

B: boolean;

D: boolean;

**begin**

writeln('Введите количество элементов массива ');

readln(n);

writeln('Введите элементы массива:');

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

**begin**

read(arr[i]);

**end**;

B := true;

D := true;

**for** i := 2 **to** n **do**

**begin**

**if** i **mod** 2 = 0 **then**

**begin**

**if** (i < n) **and** (arr[i] < arr[i + 2]) **then**

D := false;

**if** (i < n) **and** (arr[i] > arr[i + 2]) **then**

B := false;

**end**;

**end**;

M := B **or** D;

**if** M **then**

writeln('Подпоследовательность элементов на четных позициях является монотонной.')

**else**

writeln('Подпоследовательность элементов на четных позициях не является монотонной.');

readln;

**end**.

2 Задание

**var**

inputStr, subStr: **array**[1..100] **of** char;

lenStr, lenSub, firstPos, lastPos, i, j: integer;

found: boolean;

**begin**

writeln('Введите строку:');

lenStr := 0;

**repeat**

lenStr := lenStr + 1;

read(inputStr[lenStr]);

**until** eoln;

readln;

writeln('Введите подстроку:');

lenSub := 0;

**repeat**

lenSub := lenSub + 1;

read(subStr[lenSub]);

**until** eoln;

readln;

firstPos := -1;

i := 1;

**while** (i <= lenStr - lenSub + 1) **and** (firstPos = -1) **do**

**begin**

found := true;

**for** j := 1 **to** lenSub **do**

**if** inputStr[i + j - 1] <> subStr[j] **then**

found := false;

**if** found **then**

firstPos := i;

i := i + 1;

**end**;

lastPos := -1;

i := lenStr - lenSub + 1;

**while** (i >= 1) **and** (lastPos = -1) **do**

**begin**

found := true;

**for** j := 1 **to** lenSub **do**

**if** inputStr[i + j - 1] <> subStr[j] **then**

found := false;

**if** found **then**

lastPos := i;

i := i - 1;

**end**;

**if** firstPos <> -1 **then**

**begin**

**for** i := firstPos **to** lenStr - lenSub **do**

inputStr[i] := inputStr[i + lenSub];

lenStr := lenStr - lenSub;

**end**;

**if** (lastPos <> -1) **and** (lastPos <> firstPos) **then**

**begin**

**for** i := lastPos **to** lenStr - lenSub **do**

inputStr[i] := inputStr[i + lenSub];

lenStr := lenStr - lenSub;

**end**;

writeln('Итоговая строка: ');

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

write(inputStr[i]);

writeln;

**end**.

1. **Результат выполнения программы:**

1 Задание – рисунок 3

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – результат выполнения программы №1

2 Задание – рисунки 4

Рисунок 4 – результат выполнения программы №2

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

1. **Вывод:**

Получение базовых навыков работы с одномерными массивами является важной целью в изучении программирования на Pascal и закладывает фундамент для дальнейшего освоения более сложных структур данных и алгоритмов. Одномерные массивы представляют собой последовательности элементов одного типа, позволяющие организовывать данные в упорядоченные коллекции. Умение работать с массивами имеет широкий спектр практических применений: от простых задач, таких как хранение и поиск значений, до более сложных, включая статистическую обработку данных и реализацию алгоритмов сортировки. Кроме того, освоение операций над массивами, таких как инициализация, доступ к элементам, модификация и перебор, развивает логическое мышление и аналитические способности, что крайне важно для любого программиста. Знание работы с одномерными массивами также создаст базу для изучения многомерных массивов, что значительно расширит горизонты возможностей при решении различных задач.