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 строки»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программированния»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Сибейкин Иван Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Цель работы

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

2.Формулировка задания (с вариантом)

Вариант 15

Написать программу, выполняющую поиск такого минимального к, что сумма первых к елементов массива превосходит деданное число s.

﻿﻿﻿Написать программу, определяющую является ли заданная строка палиндромом (без учета регистра).

﻿﻿﻿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

**Алгоритм 1 задания**

Алгоритм поиска минимального значения k, такое что сумма первых k элементов массива превышает заданное число s, начинается с ввода размера массива n и значения s. Затем создается массив для хранения n элементов. Инициализируется переменная sum, которая будет хранить текущую сумму элементов, а также переменная k, установленная на 0 для отслеживания количества обработанных элементов.

Далее последовательно вводятся элементы массива с клавиатуры. После этого начинается цикл, который будет выполняться, пока k меньше n и текущая сумма sum меньше или равна s. Внутри цикла на каждой итерации увеличивается k на 1, чтобы учесть следующий элемент, и добавляется значение k-го элемента массива к переменной sum.

После завершения цикла выполняется проверка: если значение sum больше s, выводится значение k как минимальное количество элементов, сумма которых превышает s. В противном случае отображается сообщение о том, что сумма первых k элементов не превышает s. Алгоритм эффективно определяет минимальное количество элементов, которые необходимо учитывать для превышения заданного значения, с линейной временной сложностью

**Алгоритм Задания 2**

Ввод строки:

- Запрашиваем у пользователя ввести строку и считываем символы в массив до тех пор, пока не достигнем конца строки.

Определение длины строки:

- Определяем количество введенных символов и сохраняем это значение в переменной length.

Инициализация переменной:

- Устанавливаем логическую переменную isPalindrome в True, поскольку изначально мы предполагаем, что строка может быть палиндромом.

Проверка на палиндром:

- Проводим цикл от 1 до половины длины строки (целочисленно).

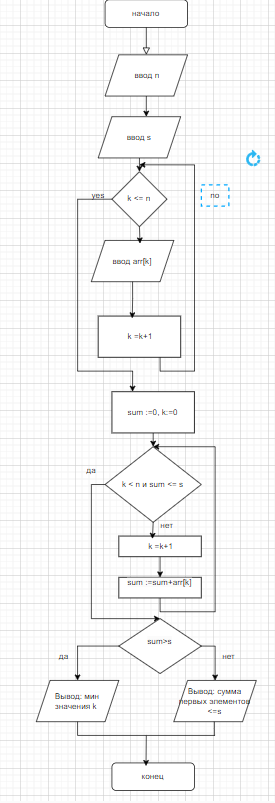
- На каждой итерации сравниваем символ из начала строки с соответствующим символом из конца строки (с использованием приведения к верхнему регистру для игнорирования регистра).

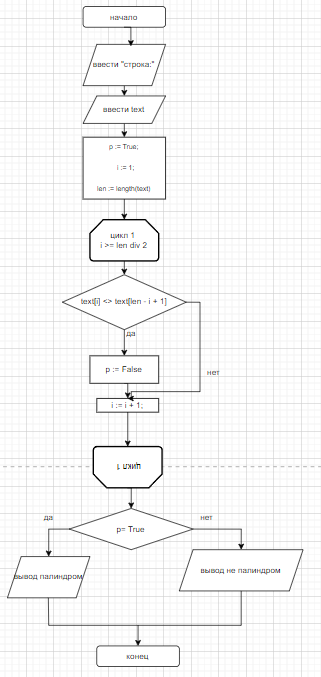
- Если символы не равны, устанавливаем isPalindrome в False и выходим из цикла.

Вывод результата:

- После завершения проверки, если isPalindrome равно True, выводим сообщение, что строка является палиндромом. В противном случае выводим сообщение, что строка не является палиндромом.

**Блок схема**





Код для программы в Pascal.

**Задание 1**

**program** vt5;

**var**

arr: **array**[1..100] **of** Integer;

n, k, sum, s: Integer;

**begin**

Write('количество элементов массива: ');

ReadLn(n);

Write('число s: ');

ReadLn(s);

WriteLn('элементы массива:');

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

**begin**

ReadLn(arr[k]);

**end**;

sum := 0;

k := 0;

**while** (k < n) **and** (sum <= s) **do**

**begin**

Inc(k);

sum := sum + arr[k];

**end**;

**if** (sum > s) **then**

WriteLn('мин значение k: ', k)

**else**

WriteLn('сумма первых элементов не превышает s.');

**end**.

**Задание 2**

**program** dkr2;

**var**

text: string;

len, i: integer;

p: boolean;

**begin**

writeln('строка:');

readln(text);

len := length(text);

p := True;

i := 1;

**while** i <= len **div** 2 **do begin**

**if** text[i] <> text[len - i + 1] **then begin**

p := False;

**end**;

i := i + 1;

**end**;

**if** p **then begin**

writeln('палиндром');

**end else begin**

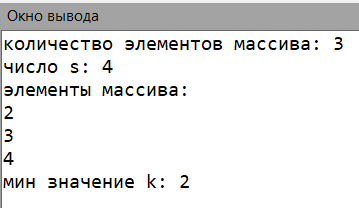
writeln('не палиндром');

**end**;

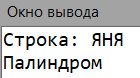
**end**.

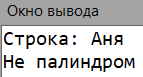
6.Результат выполнения программы

Задание 1



Задание 2





Вывод:

Изучение языка программирования Pascal позволило глубже понять принципы построения программ и эффективного применения основных языковых конструкций для решения различных задач. В ходе освоения Pascal были усвоены не только особенности синтаксиса, но и ключевые принципы программирования, что важно для развития логического мышления.

Также в процессе работы была изучена методология создания алгоритмических схем, что способствует упорядоченной организации решений и помогает лучше понять взаимодействие компонентов программы. Владение алгоритмическими методами стало основой для будущих программных разработок и создания более сложных решений.

Таким образом, поставленные цели были успешно выполнены, и приобретенные навыки будут активно применяться в последующих проектах. Базовое понимание Pascal укрепило уверенность в собственных силах и повысило мотивацию для дальнейшего изучения информатики и программирования.