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

Сунцов Александр Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

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

Формулировка задания для Варианта 16:

1. Написать программу, выполняющую поиск минимального по модулю элемента в заданном массиве.
2. Написать программу определяющую какое из слов встречается в заданной строк чаще всего.
3. Все данные вводятся с клавиатуры. При выполнении второго пункта запрещается использовать стандартные функции для работы со строками (за исключением функции определения длинны строки).

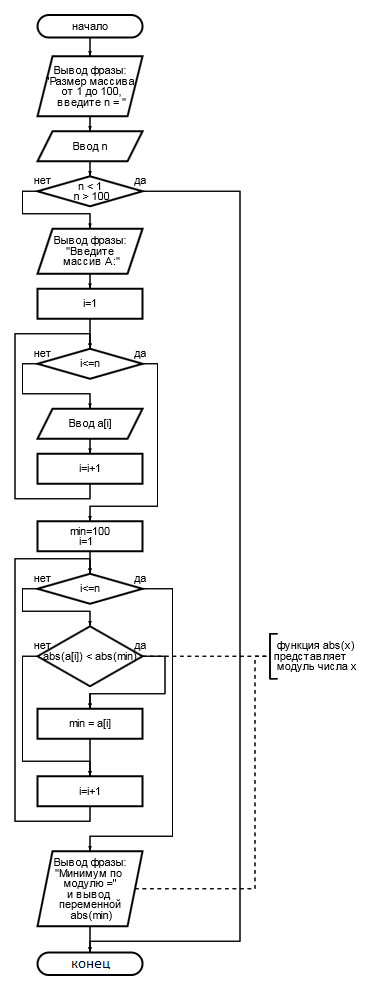


Рисунок 1 - Схема алгоритма №1

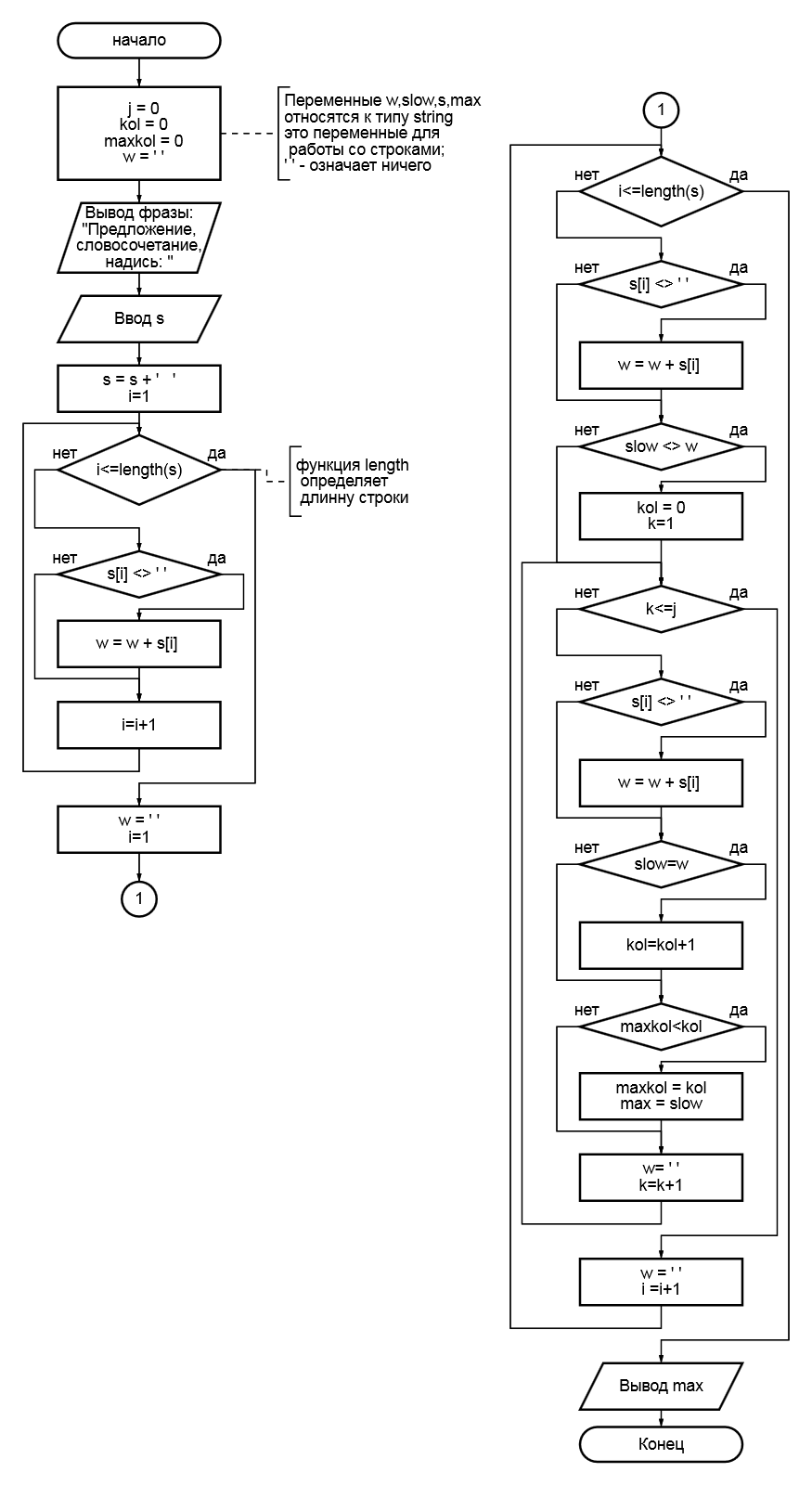


Рисунок 2 - Схема алгоритма №2

Код программы №1:

**begin**

**var** a: **array**[1..100] **of** integer;

**var** min, n: integer;

print('Размер массива от 1 до 100, введите n = ');

readln(n);

**if** (n < 1) **and** (n > 100) **then exit**;

print('Введите массив А:');

**for var** i := 1 **to** n **do** read(a[i]);

writeln;

min := 100;

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

**if** abs(a[i]) < abs(min) **then** min := a[i];

print('Минимум по модулю = ', abs(min));

**end**.

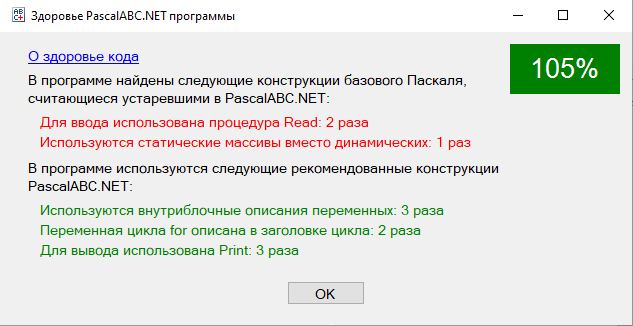


Рисунок 3 – Здоровье кода алгоритма №1

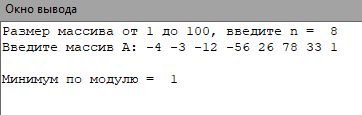


Рисунок 4 - Результат выполнения алгоритма №1

Код программы №2:

**begin**

**var** j, maxkol, kol: integer;

**var** s, w, slow, max: string;

j := 0;

kol := 0;

maxkol := 0;

w := '';

print('Предложение, словосочетание, надись: ');

readln(s);

s := s + ' ';

**for var** i := 1 **to** length(s) **do if** s[i] <> ' ' **then** w := w + s[i] **else** j := j + 1;

w := '';

**for var** i := 1 **to** length(s) **do**

**if** s[i] <> ' ' **then** w := w + s[i] **else**

**begin**

**if** slow <> w **then** kol := 0;

slow := w;

**for var** k := 1 **to** j **do**

**if** s[i] <> ' ' **then** w := w + s[i] **else**

**begin**

**if** slow = w **then** kol := kol + 1;

**if** maxkol < kol **then**

**begin**

maxkol := kol;

max := slow;

**end**;

w := '';

**end**;

w := '';

**end**;

print(max);

**end**.

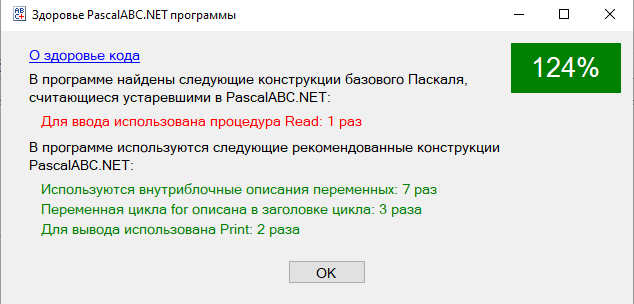


Рисунок 5 – Здоровье кода алгоритма №2

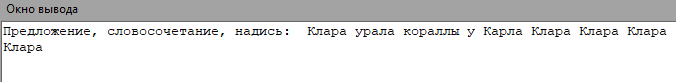


Рисунок 6 - Результат выполнения алгоритма №2

Вывод: В ходе выполнения данной контрольной работы был пройден отнюдь не легкий путь к выполнению задач, которые поставлены. Были получены знания о том, как оптимизировать код программы, чтобы “здоровье” кода в Pascal было не ниже 100%, а то и даже выше. Программы написаны без использования встроенных функций, также они оптимизированы под ввод данных пользователем. После выполнения данных задач хочется все больше изучать данную тему, чтобы достичь успехов в ней. Получены базовые навыки работы с одномерными массивами, а также освоены принципы работы со строками, как с частным случаем одномерных массивов.