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

Ветошкина Василиса Сергеевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

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

**2. Формулировка задания (Вариант:16)**

1. Написать программу, выполняющую поиск такого максимального k, что сумма первых k элементов массива не превосходит заданного числа s.

2. Написать программу, подсчитывающую число слов и букв в заданной строке.

3.Все данные вводятся с клавиатуры. При выполнении второго пункта

запрещается использовать стандартные функции для работы со строками (за

исключением функции определения длинны строки).

**3.Описание алгоритма**

***Задача № 1***

1. Введите число s.
2. Инициализируйте переменную сумма и переменную k равными 0.
3. Вводите значения элементов массива.
4. При каждой итерации: добавьте значение элемента к сумме увеличьте переменную k на 1.
5. Если сумма стала больше s: выведите предыдущее значение k (то есть k - 1).
6. Если сумма не стала больше s: выведите текущее значение k.
7. Конец программы.

***Задача № 2***

1. Запросить у пользователя ввод строки.
2. Создать переменную для подсчета количества слов и установить ее значение равным 0.
3. Создать переменную для подсчета количества букв и установить ее значение равным 0.
4. Создать переменную-флаг для определения начала нового слова и установить ее значение равным 0.
5. Итерироваться по каждому символу в строке:

* Если текущий символ является буквой, увеличить счетчик букв на 1.
* Если текущий символ является пробелом и флаг начала нового слова равен 1 (т.е. это не первый пробел подряд), увеличить счетчик слов на 1 и сбросить значение флага начала нового слова на 0.
* Если текущий символ является пробелом и флаг начала нового слова равен 0 (т.е. это первый пробел после слова), перейти к следующей итерации.
* Если текущий символ не является буквой и не является пробелом, перейти к следующей итерации.

1. Вывести результаты подсчета количества слов и букв на экран.

**4. Схема алгоритма с комментариями**

***Задача № 1***

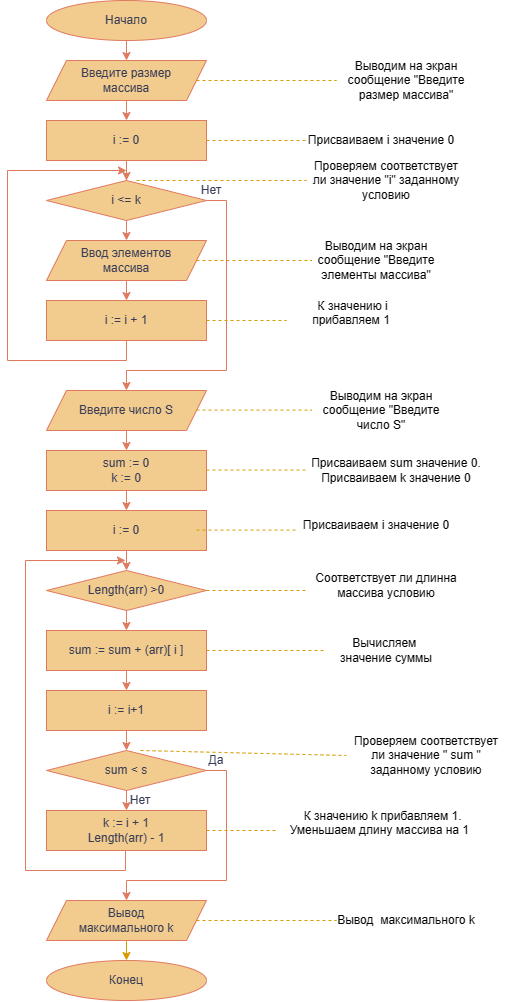


Рис 1. Схема алгоритма с комментариями к задаче № 1

***Задача № 2***

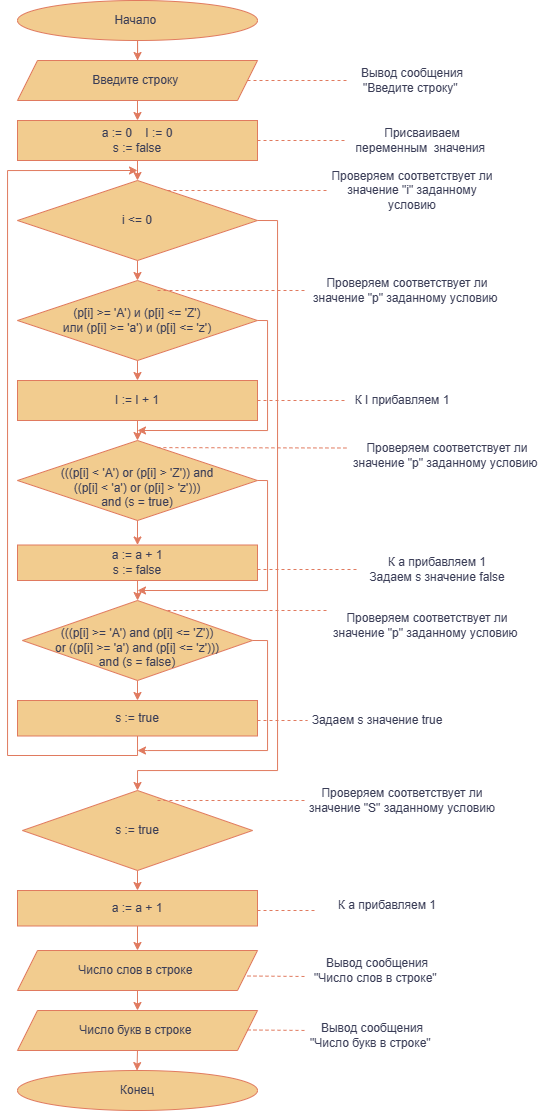


Рис 2. Схема алгоритма с комментариями к задаче № 2

**5**. **Код программы:**

***Задача № 1***

**program** z1;

**var**

arr: **array of** Integer;

s, k, sum: Integer;

i: Integer;

**begin**

Write('Введите размер массива: ');

ReadLn(k);

SetLength(arr, k);

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

**begin**

Write('Введите элемент массива: ');

ReadLn(arr[i]);

**end**;

Write('Введите число s: ');

ReadLn(s);

sum := 0;

k := 0;

**for** i := 0 **to** Length(arr) - 1 **do**

**begin**

sum := sum + arr[i];

**if** sum > s **then**

**begin**

**break**;

**end**;

k := i + 1;

**end**;

WriteLn('Максимальное k:', k);

**end**.

***Задача № 2***

**program** z2;

**var**

p: string;

a, l, i: integer;

s: boolean;

**begin**

writeln('Введите строку:');

readln(p);

a := 0;

l := 0;

s := false;

**for** i := 1 **to** Length(p) **do**

**begin**

**if** (p[i] >= 'A') **and** (p[i] <= 'Z') **or** (p[i] >= 'a') **and** (p[i] <= 'z') **then**

l := l + 1;

**if** (((p[i] < 'A') **or** (p[i] > 'Z')) **and** ((p[i] < 'a') **or** (p[i] > 'z'))) **and** (s = true) **then**

**begin**

a := a + 1;

s := false;

**end**;

**if** (((p[i] >= 'A') **and** (p[i] <= 'Z')) **or** ((p[i] >= 'a') **and** (p[i] <= 'z'))) **and** (s = false) **then**

s := true;

**end**;

**if** s = true **then**

a := a + 1;

writeln('Число слов в строке:', a);

writeln('Число букв в строке:', l);

readln;

**end**.

**6. Результат выполнения программы**

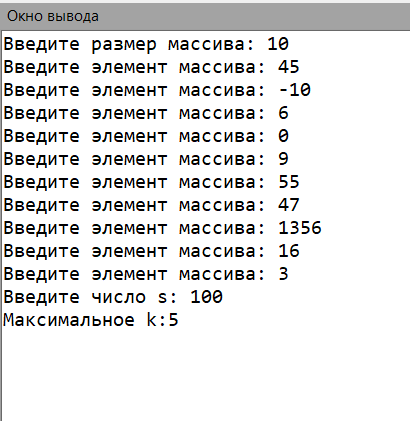


Рис 3. Результат выполнения программы задачи № 1

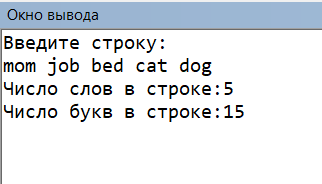
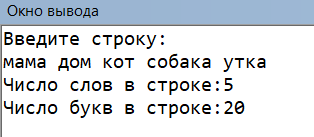


Рис 4. Результат выполнения программы задачи № 2

**7. Вывод**

Во время выполнения домашней контрольной работы мы поняли что строки и массивы являются важными и универсальными структурами данных, которые используются во множестве задач и приложений. Их использование позволяет улучшить организацию и эффективность работы с данными.

Когда мы говорим о строках, обычно мы имеем в виду отображение данных таким образом, чтобы каждый элемент массива или структуры был выведен на отдельной строке.

Это может быть полезно, когда мы хотим лучше структурировать и читать вывод программы или когда нам нужно передать эти данные в другую программу или систему.

Однако, когда мы работаем с большими объемами данных, вывод по строкам может занимать много времени и ресурсов. Поэтому в таких случаях целесообразно использовать альтернативные методы вывода данных, чтобы уменьшить их объем.

Так же в процессе выполнения работы, мы познакомились с программой для создания схем алгоритма D[raw.io](https://www.drawio.com/index.html). Draw.io — это бесплатный онлайн-сервис, который помогает создавать схемы алгоритма, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм, поэтому недавно сервис переименовали в Diagrams.net. Но старое название по-прежнему в ходу. Сервисом пользуются разные IT-специалисты — от бизнес-аналитиков до разработчиков и дизайнеров.