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-204-52-00

Беляев Макар Леонидович

Преподаватель:

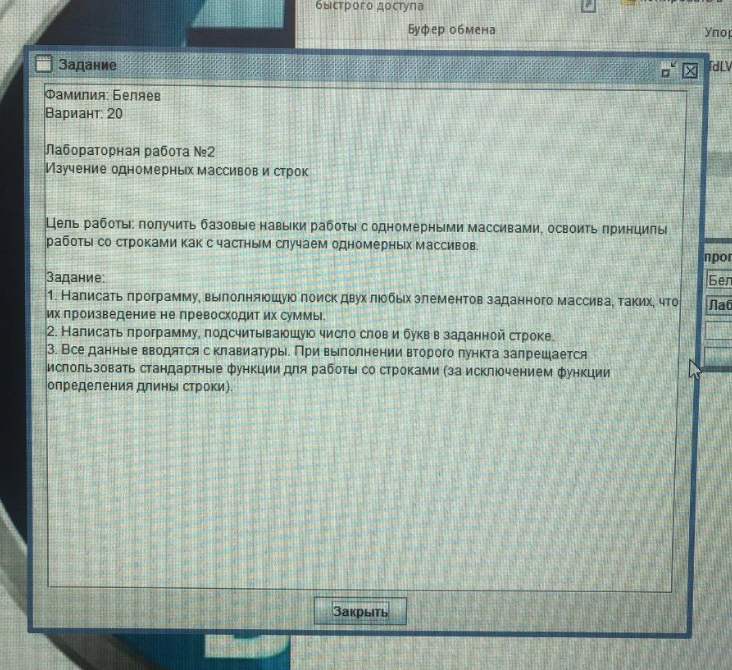
Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

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

**Формулировка задания**

**Описание алгоритма**

Данный алгоритм предназначен для поиска элементов в массиве, удовлетворяющих условию: произведение двух элементов не должно превышать их сумму.

1. Программа начинает с запроса ввода 10 целых чисел с клавиатуры для заполнения массива arr.

2. Затем программа инициализирует переменную found значением False, которая будет использоваться для отслеживания наличия элементов, удовлетворяющих условию.

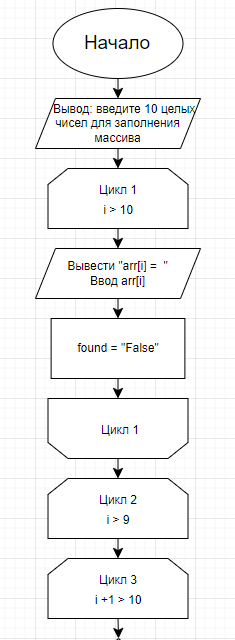
3. Далее используется вложенный цикл для перебора всех возможных пар элементов массива.

Внутри вложенного цикла происходит проверка условия: если произведение текущих элементов не превышает их сумму, выводится сообщение о найденных элементах, и переменная found устанавливается в True.

4. По завершении вложенного цикла проверяется переменная found, и в случае её значения False выводится сообщение о том, что таких элементов не было найдено.

Таким образом, алгоритм осуществляет ввод массива, а затем находит и выводит пары элементов, удовлетворяющих заданному условию.

**Схема алгоритма**

****

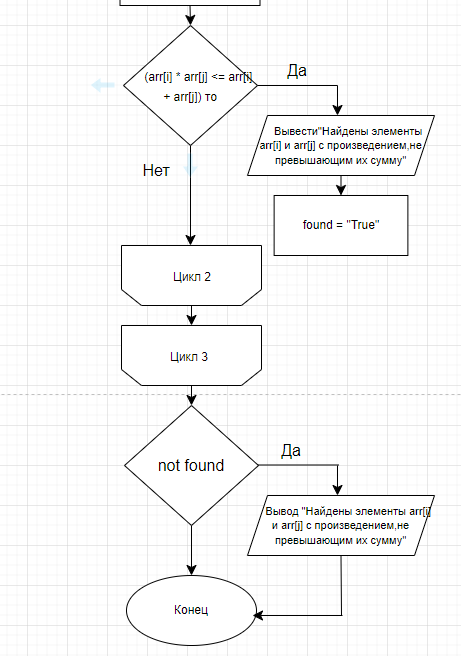
****

Рисунок 1 – Схема алгоритм задачи

**Код программы**

**program** zad1;

**var** arr: **array**[1..10] **of** integer;

i, j: integer;

found: Boolean;

**begin**

writeln('Введите 10 целых чисел для заполнения массива:');

**for** i := 1 **to** 10 **do**

**begin**

write('arr[', i, '] = ');

readln(arr[i]);

**end**;

found := False;

**for** i := 1 **to** 9 **do**

**begin**

**for** j := i + 1 **to** 10 **do**

**begin**

**if** (arr[i] \* arr[j] <= arr[i] + arr[j]) **then**

**begin**

writeln('Найдены элементы ', arr[i], ' и ', arr[j], ' с произведением, не превышающим их сумму');

found := True;

**end**;

**end**;

**end**;

**if not** found **then**

writeln('Таких элементов не найдено');

**end**.

**Результат выполнения программы**

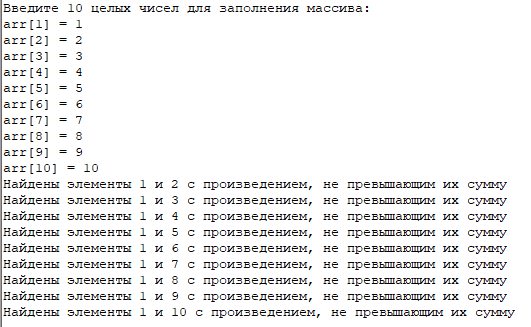
****

Рисунок 2 – Результат выполнения программы 1

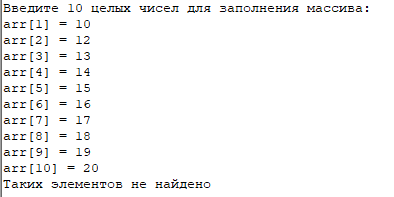
****

Рисунок 3 – Результат выполнения программы 2

**Описание алгоритма второго кода**

1. Вводится строка с клавиатуры.

2. Инициализируются счетчики слов (w) и букв (l), а также флаг (g) для проверки, была ли предыдущая проверенная буква.

3. Перебираются все символы строки в цикле от 1 до длины строки.

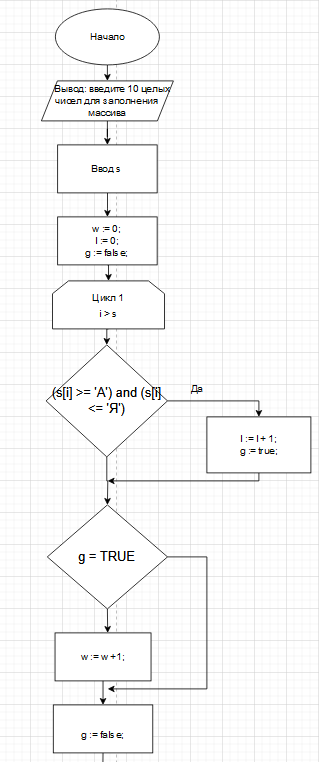
4. Если символ является буквой (проверяется по ASCII-кодам), то увеличивается счетчик букв (l) и устанавливается флаг (g) в true.

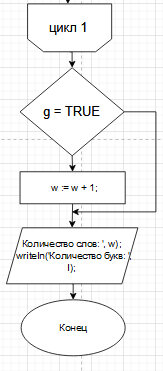
5. Если символ не является буквой, то проверяется, была ли предыдущая проверенная буква. Если была, то увеличивается счетчик слов (w). Флаг (g) устанавливается в false.

6. Если строка не заканчивается пробелом и последний символ является буквой, то увеличивается счетчик слов (w).

7. Выводятся результаты подсчета количества слов и букв в строке.

**Схема алгоритма второго кода**





**Код программы**

**program** zad2;

**var** s: string; w, l, i: integer; g: boolean;

**begin**

write('Введите строку: ');

readln(s);

w := 0;

l := 0;

g := false;

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

**begin**

**if** (s[i] >= 'А') **and** (s[i] <= 'Я') **or**

(s[i] >= 'а') **and** (s[i] <= 'я') **then**

**begin**

l := l + 1;

g := true;

**end**

**else**

**begin**

**if** g **then**

w := w + 1;

g := false;

**end**;

**end**;

**if** g **then**

w := w + 1;

writeln('Количество слов: ', w);

writeln('Количество букв: ', l);

**end**.

**Результат выполнения программы**

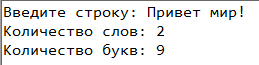


Рисунок 5 – Результат выполнения программы 1

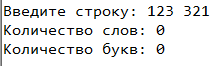


Рисунок 6 – Результат выполнения программы 2

**Вывод**

В ходе выполнения домашней лабораторной работы мы закрепили знания полученные при выполнение лабораторных работа №5-6 и №7-8. При написании кода задачи были использованы конструкции условия if изученные на лабораторных работах №1-2 и конструкции с предусловием While. Так же были использованы знания, полученные на лекциях и при выполнении лабораторных работ №5-6 и №7-8. Никаких сложностей при выполнения домашней контрольной работы не встретилось.

Так же для написания схемы алгоритма был использован интернет -ресурс «draw.io». Его использование упрощает написание схем алгоритмов, уменьшает вероятность ошибок, а также в случае пропуска какого-либо цикла его можно без каких-либо проблем вставить в любое место схемы.