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

Щепалкин Егор Васильевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы

Изучить базовую структуру организации программы и основные конструкции языка программирования Pascal.

1. Формулировка заданий
2. Написать программу, выполняющую поиск двух любых элементов заданного массива, таких, что их произведение не превосходит их суммы.
3. Написать программу, подсчитывающую число слов и букв в заданной строке.
4. Описание алгоритма №1

Алгоритм 1-ый:

1. Определяется тип переменных.
2. Вводится массив .
3. Задаётся массив, заполненный нулями.
4. Запускается цикл, который ищет пару цифр, которые удовлетворяют условие задачи.
5. Пары записываются в пустой массив и разделяются нулями.
6. На экран выводится массив с парами

Алгоритм 2-ой:

1. Определяется тип переменных.
2. Вводится строка.
3. Начинается цикл, который проводит проверку на первое слово, затем начинается основной цикл, который с помощью пробелов подсчитывает количество слов, параллельно подсчитывая все буквы.
4. На экран выводится количество слов, а под ними количество букв.
5. Схема первого алгоритма

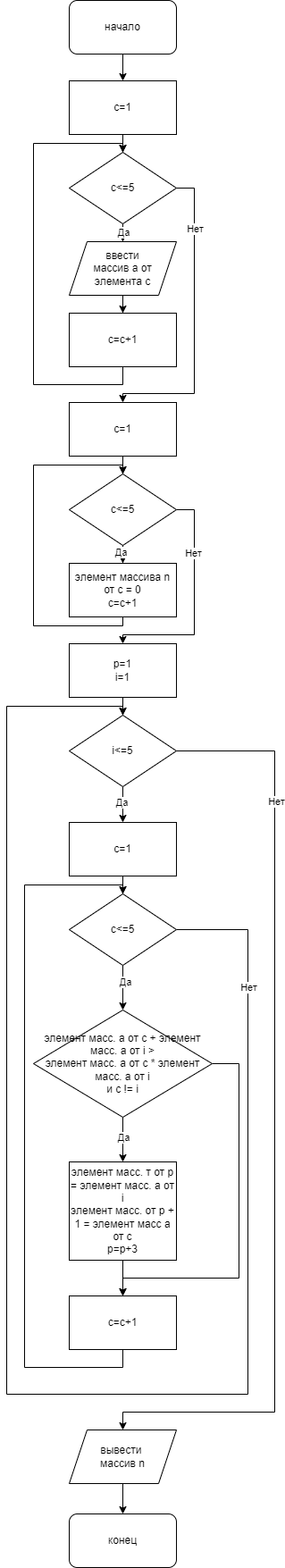
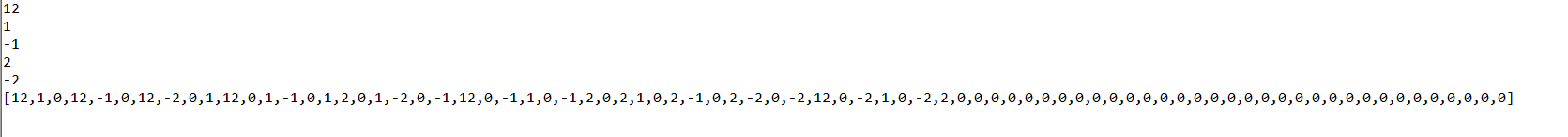


Рисунок 1 Первый алгоритм

1. Код программы для первой задачи
2. **var**
3. i,c,p:integer;
4. a:**array** [1..10] **of** integer;
5. n:**array** [1..80] **of** integer;
6. **begin**
7. **for** c:= 1 **to** 5 **do**
8. read(a[c]);
9. **for** c:= 1 **to** 80 **do**
10. n[c]:=0;
11. p:=1;
12. **for** i:=1 **to** 5 **do begin**
13. c:=1;
14. **while** c <=5 **do begin**
15. **if** (a[c]+a[i]>a[c]\*a[i]) **and** (c <> i) **then begin**
16. n[p]:=a[i];
17. n[p+1]:=a[c];
18. p:=p+3;
19. **end**;
20. c:=c+1;
21. **end**;
22. **end**;
23. writeln(n);
24. **end**.
25. Результат выполнения программы для первой задачи

Рисунок 2 Результат работы кода

1. Схема второго алгоритма

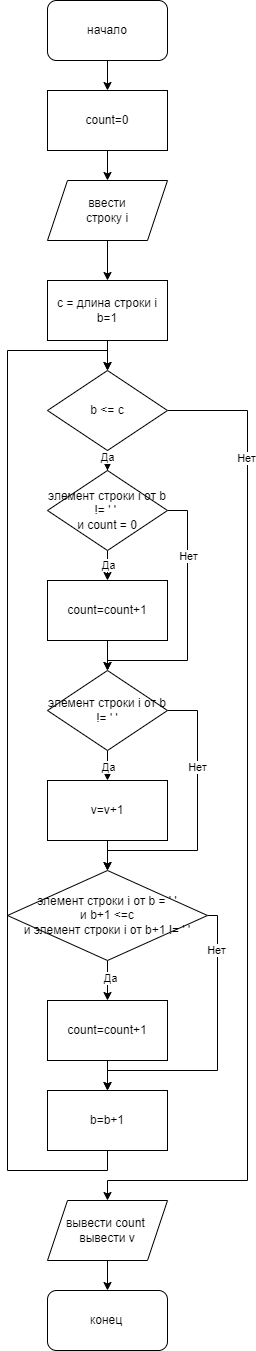


Рисунок 3 Второй алгоритм

1. Код программы для второй задачи
2. **var**
3. i:string;
4. b,c,count,v:integer;
5. **begin**
6. count:=0;
7. readln(i);
8. c:=Length(i);
9. **for** b:=1 **to** c **do begin**
10. **if** (i[b]<>' ') **and** (count = 0) **then**
11. count:= count+1;
12. **if** (i[b]<>' ') **then**
13. v:= v+1;
14. **if** (i[b]=' ') **and** (b+1<=c) **and** (i[b+1]<>' ') **then**
15. count:= count+1;
16. **end**;
17. writeln(count);
18. writeln(v);
19. **end**.
20. Результат выполнения программы для второй задачи

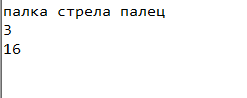


Рисунок 4 Результат работы кода

1. Вывод

В данном отчете предоставлены данные и решения задач, которые были даны в домашней контрольной работе для написания и решения на языке программирования Pascal. Для выполнения были применены знания, которые давались на лекциях и практиках по дисциплине «Основы алгоритмизации и программирования. Так же, помогли базовые и более углубленные знания в математике, в частности, в алгебре.

Во время выполнения работы были трудности с написанием кода для второй задачи. Задачу получилось решить путем применения разных идей и комбинаций, которые в конце превратились в один алгоритм.