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

Коротков Антон Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы

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

1. Формулировка задания
2. Написать программу, определяющую самый часто встречающийся элемент в заданном массиве.
3. Написать программу, определяющую какое из слов встречается в заданной строке чаще всего.
4. Описание алгоритма 1
5. Определяется тип переменных
6. Вводится значение вычисляемой переменной Х
7. В первых действиях после “if” определяется диапазон, в котором находится вычисляемая переменная Х (если переменная Х находится в диапазоне менее числа -8, вводится еще одна переменная Е)
8. После определения диапазона переменной, программа вычисляет значение функции по формуле, подходящей по условию задачи
9. После вычисления выдается результат значения функции, либо ее отсутствие
10. Так же указан шаг, для вычисления функции по условиям второй задачи
    1. Описание алгоритма 2
11. Определяется тип переменных.
12. Вводится строка.
13. Включается цикл, который отделяет слова и сортирует каждое слово по отдельности во второй цикл.
14. С помощью цикла строка переводится в массив для удобства счета.
15. Запускается ещё один цикл, который берёт по одному слову из отсортированного массива и смотрит сколько раз оно встретилось в остальном.
16. Количество появлений каждого слова строки записывается в отдельный массив.
17. Включается третий цикл, который вычисляет, какое слово встретилось больше всего раз и выводит это слово на экран.

Схема алгоритма 1

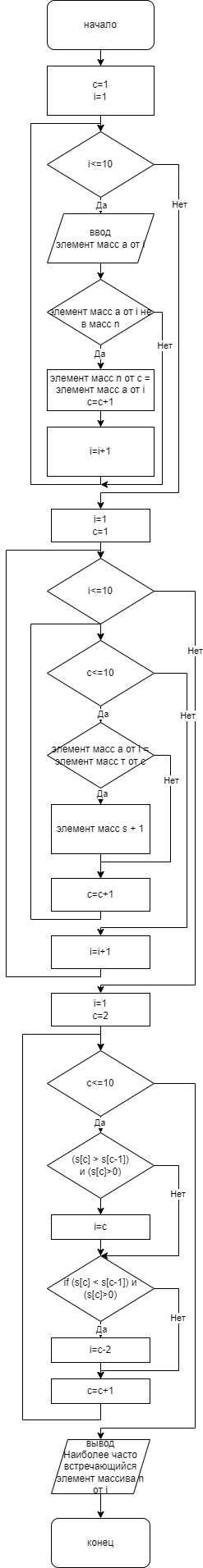


Рисунок Алгоритм задачи 1

Код программы

**var**

i,c:integer;

a,n,s:**array** [1..10] **of** integer;

**begin**

c:=1;

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

readln(a[i]);

**if** a[i] **not in** n **then begin**

n[c]:=a[i];

c:=c+1;

**end**;

**end**;

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

**for** c:=1 **to** 10 **do begin**

**if** a[i]= n[c] **then**

s[c]:=s[c]+1;

**end**;

**end**;

i:=1;

**for** c:=2 **to** 10 **do begin**

**if** (s[c] > s[c-1]) **and** (s[c]>0) **then**

i:=c;

**if** (s[c] < s[c-1]) **and** (s[c]>0) **then**

i:=c-1;

**end**;

writeln('Наиболее часто встречающийся элемент: ', n[i]);

1. **end**.

Результат выполнения программы

Задание 1.

Результаты вычислений при различных значениях массивов

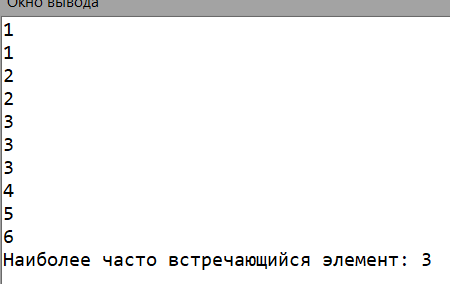


Рисунок Пример выполнения программы 1

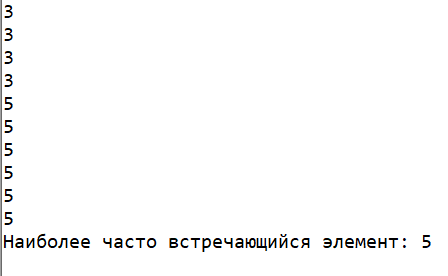
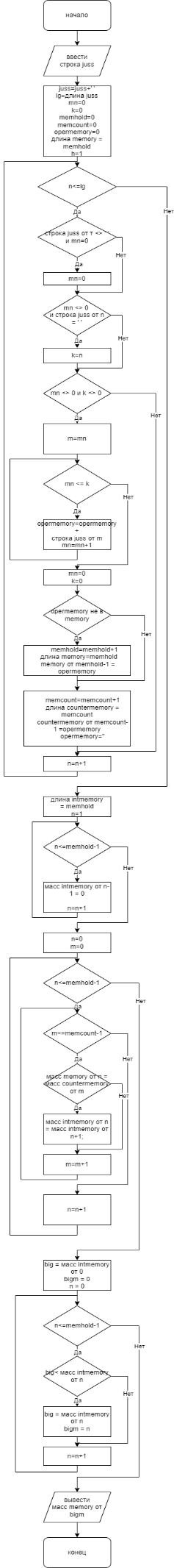


Рисунок Пример выполнения программы 1

Схема алгоритма 2



Код программы для второй задачи

**var**

juss,opermemory:string;

memory,countermemory:**array of** string;

intmemory:**array of** integer;

n,m,mn,k,lg,memhold,memcount,big,bigm:integer;

**begin**

readln(juss);

juss:=juss+' ';

lg:=Length(juss);

mn:=0;

k:=0;

memhold:=0;

memcount:=0;

opermemory:='';

SetLength(memory,memhold);

**for** n:=1 **to** lg **do begin**

**if** (juss[n]<>' ') **and** (mn=0) **then**

mn:=n;mol

**if** (mn<>0) **and** (juss[n]=' ') **then**

k:=n;

**if** (mn<>0) **and** (k<>0) **then begin**

**for** m:= mn **to** k **do begin**

opermemory:=opermemory+juss[m];

**end**;

mn:=0;

k:=0;

**if** opermemory **not in** memory **then begin**

memhold:=memhold+1;

SetLength(memory,memhold);

memory[memhold-1]:=opermemory;

**end**;

memcount:=memcount+1;

SetLength(countermemory,memcount);

countermemory[memcount-1]:=opermemory;

opermemory:='';

**end**;

**end**;

SetLength(intmemory,memhold);

**for** n:=1 **to** memhold-1 **do begin**

intmemory[n-1]:=0;

**end**;

**for** n:=0 **to** memhold-1 **do begin**

**for** m:=0 **to** memcount-1 **do begin**

**if** memory[n]=countermemory[m] **then begin**

intmemory[n]:=intmemory[n]+1;

**end**;

**end**;

**end**;

big:=intmemory[0];

bigm:=0;

**for** n:=0 **to** memhold-1 **do begin**

**if** big < intmemory[n] **then begin**

big:=intmemory[n];

bigm:=n;

**end**;

**end**;

writeln(memory[bigm]);

**end**.

1. Результат выполнения программы для второй задачи

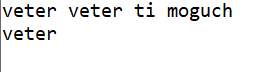


Рисунок Пример выполнения программы 2

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 решения домашней контрольной работы возникли большие трудности с решением второй задачи. Не хватало осознания итогового кода. Пришлось обратиться к коллегам и знакомым, которые смогли помочь решить задачу.