Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №4

Тема работы: Работа с одномерными массивами

Выполнил

студент: гр. 251003 Дедов Н.Ю.

Проверил: Фадеева Е.П.

Минск 2022

Содержание

[1 Постановка задачи 3](#_Toc119950722)

[1.1 Условие задачи 3](#_Toc119950723)

[1.2 Ввод данных 3](#_Toc119950724)

[1.3 Вывод данных 3](#_Toc119950725)

[2 Структура Данных 4](#_Toc119950726)

[3 Текстовый алгоритм данных 5](#_Toc119950727)

[4 Методика решения 6](#_Toc119950728)

[4.1 Условия для ввода данных 6](#_Toc119950729)

[4.2 Проверка введенных данных 6](#_Toc119950730)

[4.3 Алгоритм решения 6](#_Toc119950731)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc119950732)

[6 Результаты расчетов 8](#_Toc119950733)

[Приложение А 9](#_Toc119950734)

[Приложение Б 11](#_Toc119950735)

# Постановка задачи

## Условие задачи

Дан массив из пяти чисел а1, а2, а3, а4, а5. Среди них:

1. Если одинаковы 5, то напечатать число 1, иначе
2. Если одинаковы 4, то напечатать число 2, иначе
3. Если одинаковы 3 и 2, то напечатать число 3, иначе
4. Если одинаковы 3, то напечатать число 4, иначе
5. Если одинаковы 2 и 2, то напечатать число 5, иначе
6. Если одинаковы 2, то напечатать число 6, иначе
7. Напечатать число 7

## Ввод данных

Пользователь вводит 5 чисел.

## Вывод данных

Если все условия ввода соблюдены, то выводится число в соответствии с условием задачи.

# Структура Данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Integer | Количество чисел в массиве |
| V | Integer | Число, необходимое для цикла |
| Mas | Array [1..N] of real | Массив, содержащий числа |
| ChisNow | Real | Число, хранящее число из массива |
| I | Integer | Число, необходимое для цикла |
| PovChis | Real | Число, которое уже повторялось |
| Povtor | Boolean | Проверка есть ли хоть одно повторяющееся число |
| Schet1 | Integer | Счетчик для повторяющихся чисел |
| Schet2 | Integer | Счетчик для повторяющихся чисел |

# Текстовый алгоритм данных

Таблица 2 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  Шага | Назначение шага |
|  | N:=5 |
|  | Ввод Mas[I] |
|  | Schet1:=1 |
|  | Povtor:=False |
|  | Schet2:=0 |
|  | PovChis:=0 |
|  | Начало цикла А1. Проверка выполнение условия: ( I <= N-1). Если условие истинно, то идти к шагу 8, иначе – к шагу 20 |
|  | ChisNow:=Mas[i] |
|  | Начало цикла А2. Проверка выполнение условия: (V:= I <= N-1). Если условие истинно, то идти к шагу 10, иначе – к шагу 13 |
|  | Проверка выполнение условия: ((ChisNow=Mas[V+1]) and ((Mas[V+1]<> PovChis) or (Povtor=False))). Если условие истинно, то идти к шагу 11, иначе – к шагу 12 |
|  | Schet1:=Schet1+1 |
|  | Конец цикла А2. Вернуться к шагу 9 |
|  | Проверка выполнения условия: ((Schet2 > 0) and (Schet1 > 1)). Если условие истинно, то идти к шагу 20, иначе – к шагу 14 |
|  | Проверка выполнения условия: (Schet1 > 1). Если условие истинно, то идти к шагу 15, иначе – к шагу 19 |
|  | Schet2:=Schet1 |
|  | Schet1:=1 |
|  | Povtor:=True |
|  | PovChis:=PovNow |
|  | Конец цикла А1. Вернуться к шагу 7 |
|  | Проверка выполнения условия: (Povtor = False). Если условие истинно, то идти к шагу 21, иначе – к шагу 22 |
|  | Вывод 7. Идти к шагу 34 |
|  | Проверка выполнение условия: (Schet2 = 5). Если условие истинно, то идти к шагу 23, иначе – к шагу 24 |
|  | Вывод 1. Идти к шагу 34 |
|  | Проверка выполнение условия: (Schet2 = 4). Если условие истинно, то идти к шагу 25, иначе – к шагу 26 |
|  | Вывод 2. Идти к шагу 34 |
|  | Проверка выполнение условия: (((Schet2 = 3) and (Schet1= 2)) or ((Schet2 = 2) and (Schet1= 3))). Если условие истинно, то идти к шагу 27, иначе – к шагу 28 |
|  | Вывод 3. Идти к шагу 34 |
|  | Проверка выполнение условия: ((Schet2 = 3) and (Schet1= 1)) Если условие истинно, то идти к шагу 29, иначе – к шагу 30 |
|  | Вывод 4. Идти к шагу 34 |
|  | Проверка выполнение условия: ((Schet2 = 2) and (Schet1= 2)) Если условие истинно, то идти к шагу 31, иначе – к шагу 32 |
|  | Вывод 5. Идти к шагу 34 |
|  | Проверка выполнение условия: ((Schet2 = 2) and (Schet1= 1)) Если условие истинно, то идти к шагу 33, иначе – к шагу 34 |
|  | Вывод 6 |
|  | Останов. |

# Методика решения

## Условия для ввода данных

Fghjp[kjkl]\:

* вводимые данные должны быть числами;
* вводимые числа должны быть действительными.

## Алгоритм решения

Чтобы определить какое число вывести, мы будем идти по массиву, состоящему из чисел, введенных пользователем, с счетчиком, и будем считать количество одинаковых чисел. Если таких нет, мы переходим к следующему числу. Если же числа совпадают, то мы отмечаем это, а также в конце прохода по всему массиву, мы сохраним это число, чтобы в дальнейшем не отмечать повторяющиеся числа.

# Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Как результат выполнения программы, на экран выводятся следующие результаты расчетов:

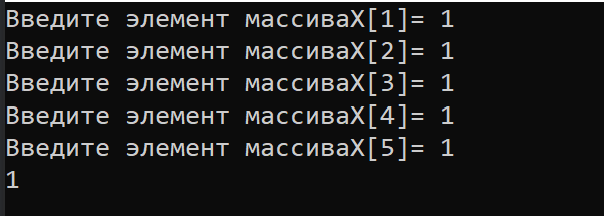


Рисунок 3 – Результаты расчетов при пяти одинаковых числах

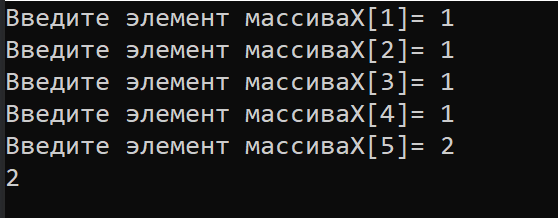


Рисунок 4 – Результаты расчетов при черырех одинаковых числах

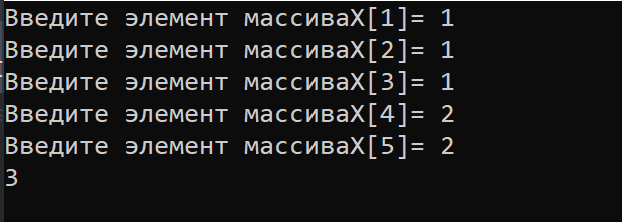


Рисунок 5 – Результаты расчетов при двух одинаковых числах и трех одинаковых числах

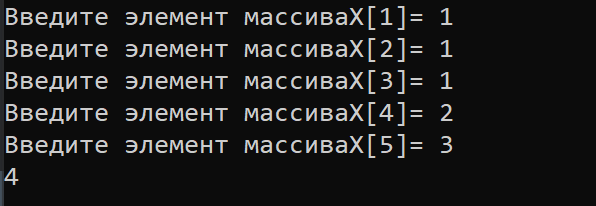


Рисунок 6 – Результаты расчетов при трех одинаковых числах

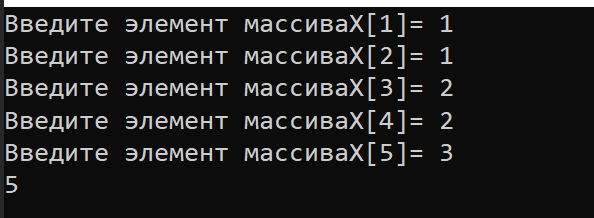


Рисунок 7 – Результаты расчетов при двух пар двух повторяющихся чисел

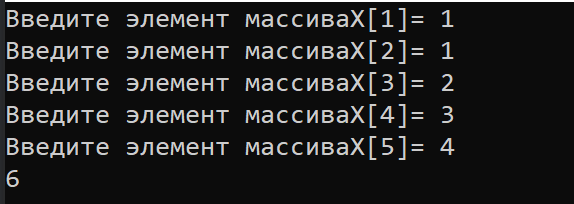


Рисунок 8 – Результаты расчетов при двух одинаковых чисел

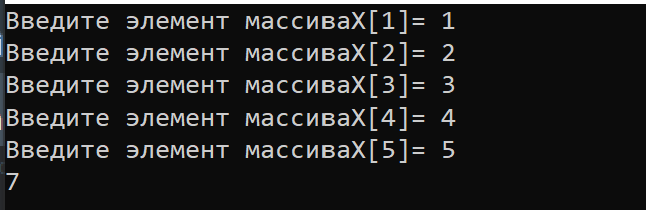


Рисунок 9 – Результаты расчетов при отсутствии повторений

Приложение А

(обязательное)

Исходный код программы

Program Lab4;

{$APPTYPE CONSOLE}

{$R \*.res}

Uses

System.SysUtils;

Const

N=5;

Var

Mas:array[1..N] of real;

I, V, Schet1, Schet2: integer;

Povtor: boolean;

ChisNow, PovChis: real;

//I – number of part of our array

//Schet1, Schet2 – for counting our identical num

//Povtor – making sure there was the same num

//ChisNow – calculating the similarity in array

//PovChis – previous number

Begin

// Input of our array

for i:=1 to N do

Begin

write('Введите элемент массива','Mas[',i,']=');

Readln(Mas[i]);

End;

Schet1:= 1;

Schet2:= 0;

Povtor:= False;

PovChis:=0;

for I := 1 to N-1 do

begin

M:=Mas[i];

for v := i to n-1 do

//Comparing this number to the whole array

if (ChisNow=Mas[v+1]) and ((Mas[v+1]<> PovChis)

or Povtor = False)) then

Schet1:=Schet1+1;

if (Schit2 > 0) and (Schit1 > 1) then

break;

if Schit1 > 1 then

begin

Schit2:=Schit1;

Schit1:=1;

Povtor:=true;

PovChis:=PovNow;

end;

end;

if Povtor = false then

writeln('7')

else

begin

if Schit2 = 5 then

writeln('1');

if Schit2 = 4 then

writeln('2');

if ((Schit2 = 3) and (Schit1 = 2)) or ((Schit1 = 3) and (Schit2 = 2))then

writeln('3');

if (Schit1=1) and (Schit2=3) then

writeln('4');

if (Schit2=2) and (Schit1=2) then

writeln('5');

if (Schit1=1) and (Schit2=2) then

writeln('6');

end;

readln(I);

end.