**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ

Департамент компьютерной инженерии

**Курс: Алгоритмизация и программирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Раздел** | **Маx**  **оценка** | **Итог.**  **Оценка**  Задание 1 | **Итог.**  **Оценка**  Задание 2 | **Итог.**  **Оценка**  Задание 3 |
| Постановка | 0,5 |  |  |  |
| Метод | 1 |  |  |  |
| Спецификация | 0,5 |  |  |  |
| Алгоритм | 1,5 |  |  |  |
| Работа программы | 1 |  |  |  |
| Листинг | 0,5 |  |  |  |
| Тесты | 1 |  |  |  |
| Вопросы | 2 |  |  |  |
| Доп. задание | 2 |  |  |  |

**ОТЧЕТ**

**по лабораторной работе № 3**

**Студент:** Камаров Лазизбек

Шухрат угли

**Группа:** БИВ201

**Вариант:** 88 (№9, №8, №5)

**Руководитель:** Ерохина Елена

Альфредовна

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**МОСКВА 2020**

Оглавление

[Задачи 2](#_Toc529909976)

[Постановка задачи 3](#_Toc529909977)

[Метод решения задачи 4](#_Toc529909978)

[Внешняя спецификация 5](#_Toc529909979)

[Алгоритм решения задачи 9](#_Toc529909980)

[Листинг программы 14](#_Toc529909981)

[Тесты 21](#_Toc529909982)

# Задачи

1. Оформить в виде функции общую часть в предложенных формулах:

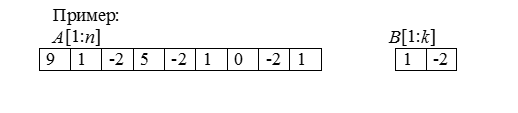
Вычислить значения по формулам:



1. Вычислить значение по формуле:
2. Дан целочисленный массив A[1:n]. Написать программу, включающую две процедуры с параметрами. В первой процедуре необходимо подсчитать количество повторений каждого элемента массива A.

Вторая процедура решает следующую задачу:

Сформировать новый массив, содержащий повторяющиеся элементы массива *A*. Элементы в новом массиве не должны повторяться.



# Постановка задачи

Дано:

1. n – цел., A[1..n]– вещ.
2. n – цел., m – цел., A[1..n,1..m] – цел.
3. n – цел., a[1..n] – цел.

Результат:

1. S, σ
2. D
3. kol [1..r], c[1..r] и b [1..k] или kol [1..r], c[1..r] и сообщение «Не удалось сформировать массив b (нет повторяющихся элементов)»

При:

1. n ∈ N, n ≤ lmax, A[i] ∈ Z (i = )
2. n ∈ N, n ≤ lmax, m ∈ N, m ≤ lmax, A[i, j] ∈ Z (i = , j = )
3. n ∈ N, n ≤ lmax, a[i, j] ∈ Z (i = , j = )

Связь:

1. см. формулу в условии
2. ∀ t = ∃ k = ∀ i = **⇒** | A [k, t] | | A [i, t] |, A [k, t] =

∃ p = : ∀ j =

1. ∀ i = ∃ z = : c[z] = a[i]

∀ m = , j = : m ≠ j, c[m] ≠ c[j]

∀ y = kol[y] = 0, ∀ L= : a[L] = c[y], kol[y] = kol[y] + 1

∃ w = : ∃ x = , a[w] = a[x] (x ≠ w) => ∃ d: b[d] = a[w] (a[x])

∀ s = , i = : s ≠ i, b[s] ≠ b[i]

# Метод решения задачи

1. См. формулу в условии
2. mах = - maxint - 1

# Внешняя спецификация

**Задача №1**

Лабораторная работа №3. Задание №1

Расчет значений по формулам

Введите количество элементов n

<n>

При err ≠ 0

Неверные данные. Повторите ввод

До err = 0

Введите элементы

<a[i]>

При err ≠ 0

Неверные данные. Повторите ввод

i = i + 1

До err = 0 и i = n

Значения S(<<S>>) , σ (<<σ >>)

**Задача №2**

Лабораторная работа №3. Задание №2

Задайте параметры матрицы В

Введите целое число n строк матрицы A 1<=n<=Lmax

<n>

При err≠0 или r<1 или r>lmax или r – вещ.

Неверные данные. Повторите ввод

До err=0 и r≥1 и r≤lmax и r – цел.

Введите целое m столбцов матрицы A 1<=m<=Lmax

<m>

При err≠0 или r<1 или r>lmax или r – вещ.

Неверные данные. Повторите ввод

До err=0 и r≥1 и r≤lmax и r – цел.

Введите <<n\*m>> элементов матрицы A по одному в каждой строке:

i =

j =

<A[i,j]>

При err≠0 или r<-maxint-1 или r>maxint или r – вещ.

Неверные данные. Повторите ввод

До err=0 и r≥-maxint-1 и r≤maxint и r – цел.

Ваша матрица A:

i =

j =

<<b[i,j]>>

D = <<d>>

**Задача №3**

Лабораторная работа №3. Задание №3

Введите длину n массива A (1<=n<=20)

<n>

При err≠0 или r<1 или r>lmax или r – вещ.

Неверные данные. Повторите ввод

До err=0 и r≥1 и r≤lmax и r – цел.

Введите элементы массива А по одному в каждой строке

i =

<a[i]>

При err≠0 или r<1 или r>lmax или r – вещ.

Неверные данные. Повторите ввод

До err=0 и r≥1 и r≤lmax и r – цел.

Ваш массив:

i =

<<a[i]>>

Повторы:

Элемент - число повторений

i =

<<c[i]>> - <<kol[i]>>

При k≠0

Сформированный массив b:

i =

<<b[k]>>

Иначе

Не удалось сформировать массив b (нет повторяющихся элементов)

# Алгоритм решения задачи

**Задача №1**

Алг «процедура startprogram»

**нач**

**вывод**('лабораторная работа №', c\_number\_lab)

**вывод**('задание ', c\_number\_zad)

**вывод**('каждое число надо вводить с новой строке(через клавишу "enter")!!!')

**кон**

Алг «функция readint»

вход min, max

**нач**

**нц**

**если** (rep = 1) **то**

**вывод**('введите целое число!!! оно не должно быть больше, чем ', lmax, '!!! ')

**все**

**ввод**(s)

rep := 1

val(s, r, err)

**кц** **пока** (err = 0) **и** (r = round(r)) **и** (r < max) **и** (r >min)

readint := round(r)

**кон**

Алг «процедура caseofelements»

**Вход** elem

**нач**

**если** ((elem mod 100 div 10 = 1) **или** (elem mod 10 = 5) **или** (elem mod 10 = 0) **или** (elem mod 10 = 6) **или** (elem mod 10 = 7) **или** (elem mod 10 = 8) **или** (elem mod 10 = 9)) **то**

**вывод**('элементов ')

**иначе**

**если** (elem mod 10 = 1) **то**

**вывод**('элемент ')

**иначе**

**вывод**('элемента ')

**все**

**все**

**кон**

Алг « процедура readmass »

**нач**

**вывод**('введите количество элементов массива от 1 **до** ', lmax,' ')

n := readint(0, lmax)

**вывод**('введите ', n, ' ')

caseofelements(n) // правильный падеж слова элемент

**вывод**('массива ')

**цикл от** i := 1 **до** n **нц** {считываем **массив**}

a[i] := readint(- maxint - 1, maxint)

**кц**

**кон**

Алг « процедура srar»

**вход** n, s, a

**переменные** i: **целые**

**нач**

**цикл от** i := 1 **до** n **нц**

s := s + a[i]

**кц** // сумма всех элементов массива

s := s / n

**кон**

Алг « процедура sigma»

**Вход** n a c

**нач**

sig := 0

**цикл от** i:= 1 **до** n **нц** // считаем сумму квадратов

sig := sig + sqr(a[i] - s)

**кц**

sig := sqrt(sig/n)

**вывод**('σ = ', sig)

**кон**

Алг « задание 1 »

**нач**

startprogram // стандартные фразы

readmass // считывание массива

srar(n, s, a)

**вывод**('s = ', s)

sigma(n, a, s)

**кон**.

**Задача №2**

Алг «процедура startprogram»

**нач**

**вывод**('лабораторная работа №', c\_number\_lab)

**вывод**('задание ', c\_number\_zad)

**вывод**('каждое число надо вводить с новой строке(через клавишу "enter")!!!')

**кон**

Алг «функция readint»

вход min, max

**нач**

**нц**

**если** (rep = 1) **то**

**вывод**('введите целое число!!! оно не должно быть больше, чем ', lmax, '!!! ')

**все**

**ввод**(s)

rep := 1

val(s, r, err)

**кц** **пока** (err = 0) **и** (r = round(r)) **и** (r < max) **и** (r >min)

readint := round(r)

**кон**

Алг «процедура caseofelements»

**Вход** elem

**нач**

**если** ((elem mod 100 div 10 = 1) **или** (elem mod 10 = 5) **или** (elem mod 10 = 0) **или** (elem mod 10 = 6) **или** (elem mod 10 = 7) **или** (elem mod 10 = 8) **или** (elem mod 10 = 9)) **то**

**вывод**('элементов ')

**иначе**

**если** (elem mod 10 = 1) **то**

**вывод**('элемент ')

**иначе**

**вывод**('элемента ')

**все**

**все**

**кон**

Алг « процедура readmatr»

**нач**

**вывод**('введите количество строчек n от 1 **до** ', lmax, ' матрицы ')

n := readint(0, lmax)// проверяем, целое **и** положительное число ли это

**вывод**('введите количество столбцов м от 1 **до** ', lmax, ' матрицы ')

m := readint(0, lmax)// проверяем, целое число ли это

**вывод**('введите ', m \* n, ' ')

caseofelements(m \* n) // правильный падеж слова элемент

**вывод**('матрицы ')

**цикл от** i := 1 **до** n **нц** {считываем матрицу}

**цикл от** j := 1 **до** m **нц**

a[i, j] := readint(- maxint - 1, maxint)

**кц**

**кц**// проверяем, целое число ли это

**кон**

Алг « процедура res»

вход n m a

**нач** // ищем максимум по модулю среди минимум = (минимум в столбце)

min:= 0//максимум по модулю должен быть минимальным по модулю

**цикл от** j := 1 **до** m **нц** // сначала обрабатываем столбцы ищем минимум

min := a[1, j] // начинаем с 1, потому что в массиве

**цикл от** i := 1 **до** n **нц** // начинаем с 1, потому что в массиве может быть одна строка

**если** **модуль**(a[i, j]) < **модуль**(min) **то**

min := a[i,j]

**кц**

**все**

**если** **модуль**(min) > **модуль**(max) **то**

max := min

**все**

**кц**

result := max

**кон**

Алг « задание 2 »

**нач**

startprogram // стандартные фразы

readmatr // считывание матрицы

**вывод**('d = ', res(n, m, a)) // ответик

**кон**.

**Задача №3**

Алг «процедура startprogram»

**нач**

**вывод**('лабораторная работа №', c\_number\_lab)

**вывод**('задание ', c\_number\_zad)

**вывод**('каждое число надо вводить с новой строке(через клавишу "enter")!!!')

**кон**

Алг «функция readint»

вход min, max

**нач**

**нц**

**если** (rep = 1) **то**

**вывод**('введите целое число!!! оно не должно быть больше, чем ', lmax, '!!! ')

**все**

**ввод**(s)

rep := 1

val(s, r, err)

**кц** **пока** (err = 0) **и** (r = round(r)) **и** (r < max) **и** (r >min)

readint := round(r)

**кон**

Алг «процедура caseofelements»

**Вход** elem

**нач**

**если** ((elem mod 100 div 10 = 1) **или** (elem mod 10 = 5) **или** (elem mod 10 = 0) **или** (elem mod 10 = 6) **или** (elem mod 10 = 7) **или** (elem mod 10 = 8) **или** (elem mod 10 = 9)) **то**

**вывод**('элементов ')

**иначе**

**если** (elem mod 10 = 1) **то**

**вывод**('элемент ')

**иначе**

**вывод**('элемента ')

**все**

**все**

**кон**

Алг « процедура readmass »

**нач**

**вывод**('введите количество элементов массива от 1 **до** ', lmax,' ')

n := readint(0, lmax)

**вывод**('введите ', n, ' ')

caseofelements(n) // правильный падеж слова элемент

**вывод**('массива ')

**цикл от** i := 1 **до** n **нц** {считываем **массив**}

a[i] := readint(- maxint - 1, maxint)

**кц**

**кон**

Алг « процедура writemass»

Вход n a

**нач**

**если** n > 0 **то**

**вывод**('сформирован **массив** ')

**цикл от** i := 1 **до** n **нц**

**вывод**(a[i]:5, ' ')

**кц**

**вывод**()

**иначе**

**вывод**('**массив** не сформирован!!! :-(')

**кон**

**все**

Алг « процедура countingrepetitioninmass »

Вход a c count na nc

**нач**

**цикл от** i := 1 **до** na **нц** // проходимся по каждому элементу массива а

j := 1

**пока** (a[i] <> c[j]){если элемент из массива а еще не записан в массив с} **и** (j <= nc){если не дошли до конца массива с} **нц** // ищем все различные значения массива а

j := j + 1

**кц**

**если** j > nc **то** //если последний элемент

nc := nc + 1 // расширяем массив с(и count)

c[nc] := a[i] // записываем этот(новый элемент) в массив с

count[nc] := 1 // записываем в массив сount(где хранится количество повторений) единицу

**иначе** // так как это не последний элемент, то это значит, что этот элемент повторился

count[j] := count[j] + 1

**все**// прибавим количество повторений

**кц**

**кон**

Алг « процедура printcount »

Вход nc c count

**нач**

**вывод**('число - количество повторений: ')

**цикл от** i := 1 **до** nc **нц**

**вывод**(c[i]:5, ' - ', count[i]:2)

**кон**

**кц**

Алг « процедура createmass »

Вход na nb a b

**нач**

nb:= 0

**цикл от** i := 1 **до** na **нц** // проходимся по каждому элементу массива а

j := 1 // начинаем проверку с самого первого элемента массива в

**пока** (a[i] <> b[j]) **и** (j <= nb) **нц** //проверяем не записано ли это значение в массиве в

j := j + 1

**кц** // переходим к следующему элементу в массиве в

**если** j > nb **то** // **если** подошли к концу массива в, но так **и** не встретили такого же значения(нет в массиве в)

// проверяем, повторяется ли этот элемент в массиве а

j := 1 // так как мы прошлись по массиву **и** переменная j нам больше не нужна, мы можем использовать её туть)

**пока** (j <= na{не дошли **до** конца массива}) **и** ((a[i] <> a[j]{**пока** элементы не повторяются}) **или** (i = j{исключаем случай, когда переменнная натыкается сама на себя})) **нц**

j := j + 1

**кц**

**если** j <= na **то** // **если** не дошли **до** конца(**то** есть наши повторяющийся элемент)

nb := nb + 1 //добавляем количество элементов в масиве b

b[nb] := a[i] // добавляем это новое значение в **массив** в

**все**

**все**

**кц**

**кон**

Алг « задание 3 »

**нач**

startprogram // стандартные фразы

readmass(na, a) // считывание массива (с проверками **и** всякими штучками)

countingrepetitioninmass(a, c, count, na, nc) //подсчет повторений значений

writemass(na, a) // **вывод** массива а

printcount(nc, c, count) // **вывод** повторений значений

createmass(na, nb, a, b) //создание нового массива в, в котором содержатся повторяющиеся элементы массива а, но без повторений

writemass(nb, b) // **вывод** массива в

**кон**.

# Листинг программы

**Задача №1**

**program** Lab3Zad1;

**const**

c\_number\_lab = 3; // номер лабы

c\_number\_zad = 1; // номер задания

lmax = 969; // максимальная длина строки/столбца

**type**

mass = **array**[1..lmax] **of** integer; // массив

**var**///// начало проги

a{массив}: mass;

n{количество строк в массиве}: integer;

s{среднее арифмитическое}: real;

**procedure** startProgram;

**begin**

writeln('Лабораторная работа №', c\_number\_lab);

writeln('Задание ', c\_number\_zad);

writeln('Каждое число надо вводить с новой строки(через клавишу "enter")');

**end**;

**function** readint(min, max: real): integer; // проверяем, целое число ли это

**var**

s: string;

r: real;

err, rep: integer;

**begin**

**repeat**

**if** (rep = 1) **then** {если эта штука уже ранее вызывалась(если пользователь ввел не число), то покажется надпись которая повторит условие}

writeln('Введите целое число. Оно не должно быть больше, чем ', lmax);

Readln(s);

rep := 1;

val(s, r, err);

**until** (err = 0) **and** (r = round(r)) **and** (r < max) **and** (r > min);

readint := round(r);

**end**;

**procedure** caseOfElements(elem: integer);

**begin**

**if** ((elem **mod** 100 **div** 10 = 1) **or** (elem **mod** 10 = 5) **or** (elem **mod** 10 = 0) **or** (elem **mod** 10 = 6) **or** (elem **mod** 10 = 7) **or** (elem **mod** 10 = 8) **or** (elem **mod** 10 = 9)) **then**

write('элементов ')

**else**

**if** (elem **mod** 10 = 1) **then**

write('элемент ')

**else**

write('элемента ')

**end**;

**procedure** readMass; // считывание массива (с проверками и всякими вещами)

**var** i, j: integer;

**begin**

write('Введите количество элементов массива от 1 до ', lmax,' ');

n := readint(0, lmax);

write('введите ', n, ' ');

caseOfElements(n);

writeln('массива ');

**for** i := 1 **to** n **do** //считываем массив

a[i] := readint(- maxint - 1, maxint);

**end**;

**procedure** SrAr(**var** n: integer; **var** s: real; **var** a:mass); // среднее арифметическое

**var** i: integer;

**begin**

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

s := s + a[i]; // сумма всех элементов массива

s := s / n;

writeln('S = ', s);

**end**;

**procedure** Sigma(n: integer; **var** a:mass; **var** s: real);

**var** i: integer; sig: real;

**begin**

sig := 0;

**for** i:= 1 **to** n **do** // считаем сумму квадратов

sig := sig + sqr(a[i] - S);

sig := sqrt(sig/n); // извлекаем корень из sig поделенного на n(количество элементов)

writeln('σ = ', sig)

**end**;

**begin**

startProgram; // стандартные фразы

readMass; // считывание массива (с проверками и всякими вещами)

SrAr(n, s, a); //среднее арифмитическое

Sigma(n, a, s); //сигма

**end**.

**Задача №2**

**program** Lab3Zad2;

**const**

c\_number\_lab = 3;

c\_number\_zad = 2;

lmax = 966; // максимальная длина строки/столбца

**type**

matr = **array** [1..lmax, 1..lmax] **of** integer; // матрица

**var**

a{матрица}: matr;

n{количество строк в матрице}, m{количество столбцов в матрице}: integer;

**procedure** startProgram; // стандартные фразы

**begin**

writeln('Лабораторная работа №', c\_number\_lab);

writeln('Задание ', c\_number\_zad);

writeln('Каждое число надо вводить с новой строки(через клавишу "enter")');

**end**;

**function** readint(min, max: real): integer; // проверяем, целое число ли это

**var**

s: string;

r: real;

err, rep: integer;

**begin**

**repeat**

**if** (rep = 1) **then** {если эта штука уже ранее вызывалась(если чел ввел не число), то покажется надпись которая повторит условие}

writeln('Введите целое число. Оно не должно быть больше, чем ', lmax);

Readln(s);

rep := 1;

val(s, r, err);

**until** (err = 0) **and** (r = round(r)) **and** (r < max) **and** (r > min);

readint := round(r);

**end**;

**procedure** caseOfElements(elem: integer);

**begin**

**if** ((elem **mod** 100 **div** 10 = 1) **or** (elem **mod** 10 = 5) **or** (elem **mod** 10 = 0) **or** (elem **mod** 10 = 6) **or** (elem **mod** 10 = 7) **or** (elem **mod** 10 = 8) **or** (elem **mod** 10 = 9)) **then**

write('элементов ')

**else**

**if** (elem **mod** 10 = 1) **then**

write('элемент ')

**else**

write('элемента ')

**end**;

**procedure** readMatr; // считывание матрицы (с проверками и всякими вещами)

**var** i, j: integer;

**begin**

write('Введите количество строчек N от 1 до ', lmax, ' матрицы ');

n := readint(0, lmax);// проверяем, целое и положительное число ли это

write('Введите количество столбцов М от 1 до ', lmax, ' матрицы ');

m := readint(0, lmax);// проверяем, целое число ли это

write('введите ', m \* n, ' ');

caseOfElements(m \* n); // правильный падеж слова элемент

writeln('матрицы ');

**for** i := 1 **to** n **do** {считываем матрицу}

**for** j := 1 **to** m **do**

a[i, j] := readint(- maxint - 1, maxint);// проверяем, целое число ли это

**end**;

**function** res(n, m: integer; **var** a:matr) : integer;

**var** min, max, i, j: integer;

**begin** // ищем максимум по модулю среди min = (минимум в столбце)

max := 0;//максимум по модулю должен быть минимальным по модулю

**for** j := 1 **to** m **do** // сначала обрабатываем столбцы ищем min

**begin**

min := a[1, j]; // начинаем с 1, потому что в массиве

**for** i := 1 **to** n **do** // начинаем с 1, потому что в массиве может быть одна строка

**if** abs(a[i, j]) < abs(min) **then**

min := a[i,j]; //ищем min

**if** abs(min) > abs(max) **then**

max := min; //сравниваем наш максимум с минимумом(не больше ли минимум максимума)

**end**;

result := max;

**end**;

**begin**

startProgram; // стандартные фразы

readMatr; // считывание матрицы

writeln('D = ', res(n, m, a));

**end**.

**Задача №3**

**program** Lab3Zad3;

**const**

c\_number\_lab = 3;

c\_number\_zad = 3;

lmax = 696;// максимальная длина строки/столбца

**type**

mass = **array**[1..lmax] **of** integer;// массив

matr = **array** [1..lmax, 1..lmax] **of** integer;// матрица

**var**// начало проги, описание переменных

a{масссив 1 (исходный)}, b{массив 2 (будем его создавать)}, c{копия массива 1, но без повторений}, count{количество повторений каждого элемента массива 3(с)}: mass;

na{количество элементов в массиве A}, nb{количество элементов в массиве B}, nc{количество элементов в массиве B}: integer;

**procedure** startProgram;// стандартные фразы

**begin**

writeln('Лабораторная работа №', c\_number\_lab);

writeln('Задание ', c\_number\_zad);

writeln('Каждое число надо вводить с новой строки(через клавишу "enter")');

**end**;

**function** readint(min, max: real): integer;// проверяем, целое число ли это

**var**

s: string;

r: real;

err, rep: integer;

**begin**

**repeat**

**if** (rep = 1) **then** {если эта штука уже ранее вызывалась(если чел ввел не число), то покажется надпись которая повторит условие}

writeln('Введите целое число. Оно не должно быть больше, чем ', lmax);

Readln(s);

rep := 1;

val(s, r, err);

**until** (err = 0) **and** (r = round(r)) **and** (r < max) **and** (r > min);

readint := round(r);

**end**;

**procedure** caseOfElements(elem: integer);

**begin**

**if** ((elem **mod** 100 **div** 10 = 1) **or** (elem **mod** 10 = 5) **or** (elem **mod** 10 = 0) **or** (elem **mod** 10 = 6) **or** (elem **mod** 10 = 7) **or** (elem **mod** 10 = 8) **or** (elem **mod** 10 = 9)) **then**

write('элементов ')

**else**

**if** (elem **mod** 10 = 1) **then**

write('элемент ')

**else**

write('элемента ')

**end**;

**procedure** readMass(**var** n: integer; **var** a: mass);// считывание массива (с проверками и всякими штучками)

**var**

i: integer;

**begin**

write('Введите количество элементов массива от 1 до ', lmax, ': ');

n := readint(0, lmax);

write('Введите ', n, ' ');

caseOfElements(n);

writeln('массива ');

**for** i := 1 **to** n **do** {считываем массив}

a[i] := readint(-maxint - 1, maxint);

**end**;

**procedure** writeMass(**var** n: integer; **var** a: mass);

**var**

i: integer;

**begin**

**if** n > 0 **then**

**begin**

writeln('Сформирован массив ');

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

write(a[i]:5, ' ');

writeln();

**end**

**else**

writeln('Массив не сформирован =(')

**end**;

**procedure** countingRepetitionInMass(**var** a, c, count: mass; **var** na, nc: integer);//подсчет повторений каждого значения массива А и запись разных чисел в массив С и запись количества повторений в массив count

**var**

i, j: integer;

**begin**

**for** i := 1 **to** na **do** // проходимся по каждому элементу массива А

**begin**

j := 1;

**while** (a[i] <> c[j]){если элемент из массива А еще не записан в массив С} **and** (j <= nc){если не дошли до конца массива С} **do** // ищем все различные значения массива А

j := j + 1;

**if** j > nc **then** //если последний элемент

**begin**

nc := nc + 1; // расширяем массив С(и Count)

c[nc] := a[i]; // записываем этот(новый элемент) в массив С

count[nc] := 1; // записываем в массив Сount(где хранится количество повторений) единицу

**end**

**else** // так как это не последний элемент, то это значит, что этот элемент повторился

count[j] := count[j] + 1;// прибавим количество повторений

**end**;

**end**;

**procedure** printCount(**var** nc: integer; **var** c, count: mass);

**var**

i: integer;

**begin**

writeln('Число - количество повторений: ');

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

writeln(c[i]:5, ' - ', count[i]:2)

**end**;

**procedure** createMass(**var** na, nb: integer; **var** a, b: mass);//создание нового массива В, в котором содержатся повторяющиеся элементы массива А, но без повторений

**var**

i, j: integer;

**begin**

nb:= 0;

**for** i := 1 **to** na **do** // проходимся по каждому элементу массива А

**begin**

j := 1; // начинаем проверку с самого первого элемента массива В

**while** (a[i] <> b[j]) **and** (j <= nb) **do** //проверяем не записано ли это значение в массиве В

j := j + 1; // переходим к следующему элементу в массиве В

**if** j > nb **then** // если подошли к концу массива В, но так и не встретили такого же значения(нет в массиве В)

**begin** // проверяем, повторяется ли этот элемент в массиве А

j := 1; // так как мы прошлись по массиву и переменная j нам больше не нужна, мы можем использовать её туть)

**while** (j <= na{не дошли до конца массива}) **and** ((a[i] <> a[j]{пока элементы не повторяются}) **or** (i = j{исключаем случай, когда переменнная натыкается сама на себя})) **do**

j := j + 1;

**if** j <= na **then** // если не дошли до конца(то есть наши повторяющийся элемент)

**begin**

nb := nb + 1; //добавляем количество элементов в масиве B

b[nb] := a[i]; // добавляем это новое значение в массив В

**end**;

**end**;

**end**;

**end**;

**begin**

startProgram; // стандартные фразы

readMass(na, a); // считывание массива (с проверками и всякими вещами)

countingRepetitionInMass(a, c, count, na, nc); //подсчет повторений значений

writeMass(na, a); // вывод массива А

printCount(nc, c, count); // вывод повторений значений

createMass(na, nb, a, b); //создание нового массива В, в котором содержатся повторяющиеся элементы массива А, но без повторений

writeMass(nb, b); // вывод массива В

**end**.

# Тесты

**Задача №1**

|  |  |
| --- | --- |
| Входные данные: | Результат: |
| n = 5  A = [1, 1, 2, 2, 5] | S = 2.2  σ = 1.46969384566991 |
| n = 3  A = [1, 2, 3] | S = 2  σ = 0.816496580927726 |

**Задача №2**

|  |  |
| --- | --- |
| Входные данные: | Результат: |
| n = 3, m = 3  1 2 3  B = 4 5 6  7 8 9 | D = 3 |

**Задача №3**

|  |  |
| --- | --- |
| Входные данные: | Результат: |
| n = 7  A = [4, 5, 9, 8, 7, 5, 4] | Сформирован массив  4 5 9 8 7 5 4  Число - количество повторений:  4 - 2  5 - 2  9 - 1  8 - 1  7 - 1  Сформирован массив  4 5 |
| n = 8  A = [9, 1, 8, 2, 7, 3, 6, 4] | Сформирован массив  9 1 5 2 7 3 6 4  Число - количество повторений:  9 - 1  1 - 1  5 - 1  2 - 1  7 - 1  3 - 1  6 - 1  4 - 1  Массив не сформирован =( |