**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**Составление программ с использованием массивов**

**в качестве формальных параметров подпрограмм**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 А.В. Радионов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

Доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Цель:** научиться составлять программы в Pascal с использованием массивов в качестве формальных параметров подпрограмм (процедур и функций).

**Теоретическая часть.**

*Вспомним:*

**Подпрограмма** - это фрагмент кода, который имеет свое имя и создается в случае необходимости выполнять этот код несколько раз. Подпрограммы не необходимы, но их наличие заметно облегчает работу программиста и увеличивает «ценность» кода.

**Массив** - это определенное количество элементов, относящихся к одному и тому же типу данных, которые имеют одно имя, и каждый элемент имеет свой индекс - порядковый номер.

*Подпрограммы (процедуры и функции)* зачастую в качестве входных параметров должны обрабатывать массивы данных.   
  
В простейшем случае размерность обрабатываемых массивов постоянна, их можно описать в вызывающем блоке как отдельный тип данных, в котором указан как тип элементов, так и их число. Этот тип данных должен быть использован в заголовке подпрограммы. Перед ее вызовом во внешнем блоке программы должен быть сформирован статический массив данного типа, который передается в подпрограмму в качестве формального параметра. Такое способ назовем передачей с объявлением (описанием) типа массива в вызывающем блоке.

Зачастую подпрограмма-процедура или функция должна обработать некоторый массив данных. Однако структурный тип массива нельзя непосредственно указать в списке формальных параметров. В этом случае тип массива объявляется заранее в разделе описаний с помощью подраздела *type.*

**Массивы, как формальные параметры подпрограмм:**

**type**

matrix = **array** [1..n, 1..m] **of** real;

arr = **array** [1..n] **of** real;

**procedure** < имя процедуры > (a: matrix; **var** b: matrix);

// раздел описаний процедуры

**begin**

// исполняемая часть процедуры

**end**;

**function** < имя функции > (b: arr): real;

// раздел описаний функции

**begin**

// исполняемая часть функции

**end**;

*С динамическим массивом:*

**type**

matrix = **array of array of** real;

arr = **array of** real;

**procedure** < имя процедуры > (a: matrix; **var** b: matrix);

// раздел описаний процедуры

**begin**

// исполняемая часть процедуры

**end**;

**function** < имя функции > (b: arr): real;

// раздел описаний функции

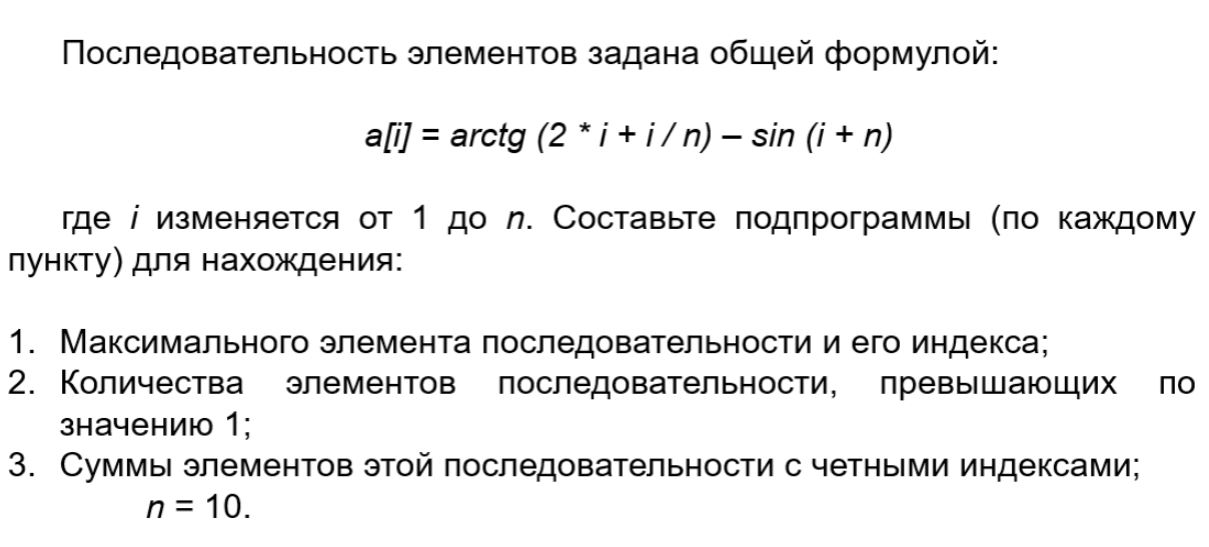
**begin**

// исполняемая часть функции

**end**;

**Практическая часть.**

**Задание 1**

****

**Программная реализация:**

**program** lb\_8\_pr\_1;

**type**

arr = **array**[1..10] **of** real;

**var**

b: arr;

i: integer;

n := 10;

**function** max(a: arr): integer;

**var**

i: integer;

max: real;

**begin**

max := a[1];

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

**if** a[i] > max **then**

**begin**

result := i;

max := a[i];

**end**;

**end**;

**function** posl(a: arr): real;

**var**

i: integer;

**begin**

result := 0;

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

**if** a[i] > 1 **then** result := result + 1;

**end**;

**function** summ(a: arr): real;

**var**

i: integer;

**begin**

result := 0;

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

**if** i **mod** 2 = 0 **then** result := result + a[i];

**end**;

**begin**

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

**begin**

b[i] := arctan(2 \* i + i / n) - sin(i + n);

writeln(b[i]);

**end**;

writeln;

writeln('Максимальный элемент последовательности - ', b[max(b)], ' и его индекс - ', max(b));

writeln('Количество элементов последовательности, превышающих по значению 1 - ', posl(b));

writeln('Сумма элементов этой последовательности с четными индексами: ', summ(b));

**end**.

**Ответ:**

2.1263673234445

1.87362606392643

0.993212417241665

0.461699011941889

0.825556780295023

1.77950057249051

2.46427125392093

2.26232991425792

1.36805835488943

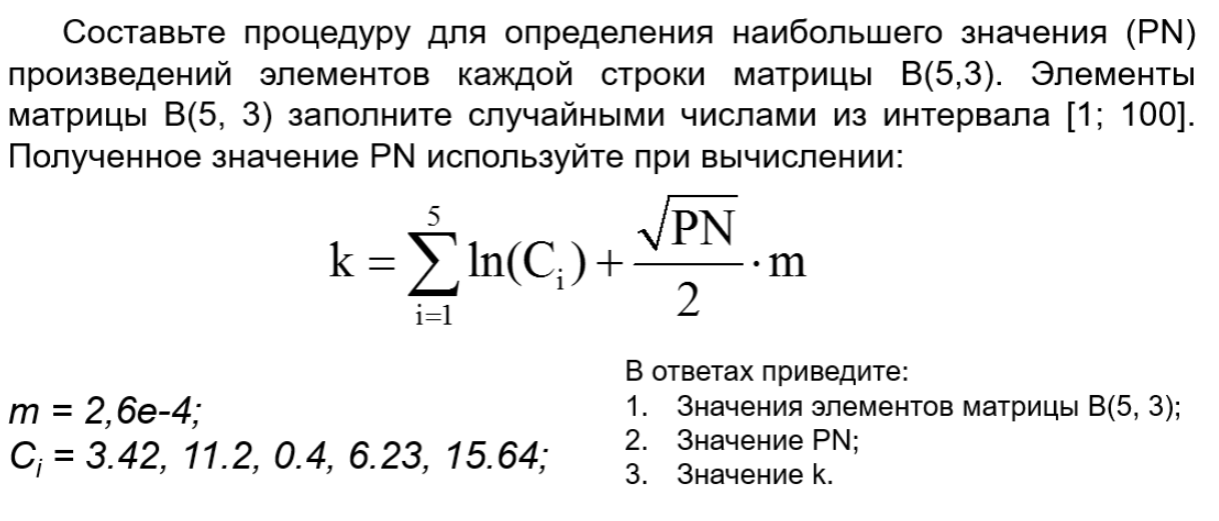
0.610267972790286

Максимальный элемент последовательности - 2.46427125392093 и его индекс - 7

Количество элементов последовательности, превышающих по значению 1 - 6

Сумма элементов этой последовательности с четными индексами: 6.98742353540703

**Задание 2**

****

**Программная реализация:**

**program** lb\_8\_pr\_2;

**type**

arr = **array**[1..5, 1..3] **of** real;

**var**

c: **array** [1..5] **of** real = (3.42, 11.2, 0.4, 6.23, 15.64);

**var**

b: arr;

i, j: integer;

PN, l, h, k: real;

m := 0.00026;

**procedure** f(**var** b: arr; **var** PN: real; **var** l: real);

**var**

i, j: integer;

**var**

max: real;

**begin**

max := 1;

PN := 0;

**for** i := 1 **to** 5 **do**

**begin**

**for** j := 1 **to** 3 **do**

max := b[i, j] \* max;

**if** max > PN **then**

**begin**

PN := max;

l := i;

**end**;

max := 1;

**end**;

**end**;

**begin**

**for** i := 1 **to** 5 **do**

**begin**

**for** j := 1 **to** 3 **do**

**begin**

b[i, j] := random(100) + 1;

write(b[i, j]:5);

**end**;

writeln;

**end**;

f(b, PN, l);

writeln;

writeln('Наибольшее значение произведений элементов у ', l, ' строки матрицы В = ', PN);

**for** i := 1 **to** 5 **do**

h := h + ln(c[i]);

k := h + sqrt(PN) \* m / 2;

writeln('k = ', k);

**end**.

**Ответ:**

89 98 60

53 20 20

49 96 96

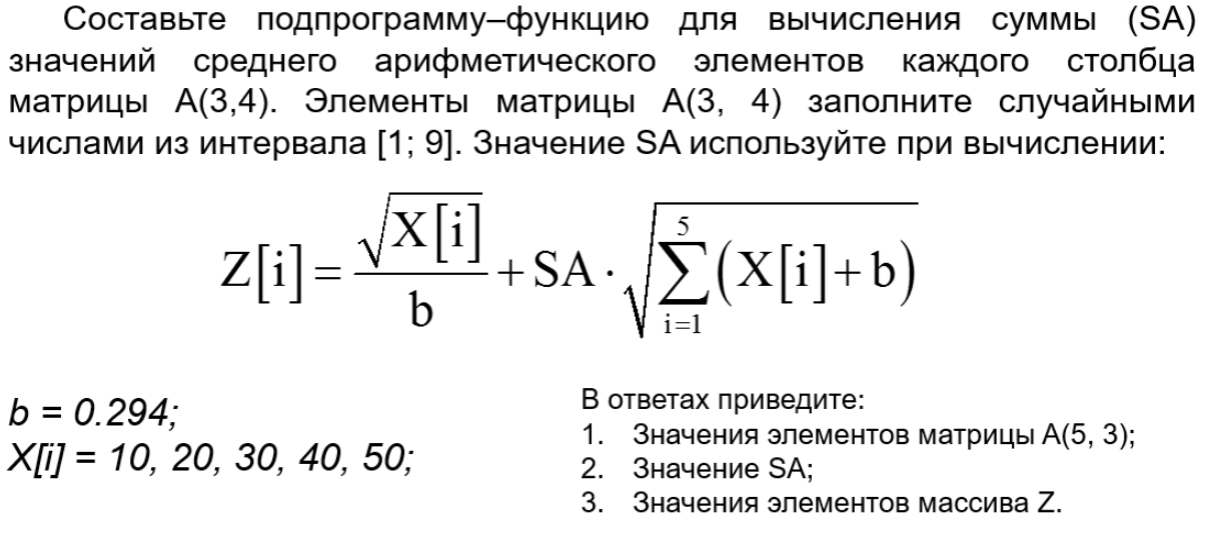
93 1 62

26 49 12

Наибольшее значение произведений элементов у 1 строки матрицы В = 523320

k = 7.40251478318846

**Задание 3**

****

**Программная реализация:**

**program** lb\_8\_pr\_3;

**type**

arr = **array**[1..3, 1..4] **of** real;

**var**

x: **array** [1..5] **of** real = (10, 20, 30, 40, 50);

**var**

z: **array** [1..5] **of** real;

i, j: integer;

a: arr;

summ: real;

b := 0.294;

**function** SA(n: arr): real;

**var**

i, j: integer;

s: real;

**begin**

**for** j := 1 **to** 4 **do**

**begin**

s := 0;

**for** i := 1 **to** 3 **do**

s := s + n[i, j];

result := result + s / 3;

**end**;

**end**;

**begin**

writeln('Значения элементов матрицы А:');

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 4 **do**

**begin**

a[i, j] := random(9) + 1;

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

**end**;

writeln;

**end**;

writeln;

writeln('Сумма значений среднего арифметического элементов каждого столбца матрицы А = ', SA(a));

**for** i := 1 **to** 5 **do**

summ := summ + x[i] + b;

writeln;

writeln('Значения элементов матрицы Z:');

**for** i := 1 **to** 5 **do**

**begin**

z[i] := (sqrt(x[i]) / b) + SA(a) \* sqrt(summ);

write(z[i]:9:2);

**end**;

**end**.

**Ответ:**

Значения элементов матрицы А:

2 2 4 8

5 4 7 2

6 2 5 4

Сумма значений среднего арифметического элементов каждого столбца матрицы А = 17

Значения элементов матрицы Z:

219.98 224.44 227.85 230.74 233.28

**Выводы:**

В ходе работы мною изучены методы использования массивов в качестве формальных параметров подпрограмм, которые были отработаны на практике - написаны программы на языке Паскаль.