**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 как формальных параметров подпрограмм**

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

Выполнил студент гр. А.В. Исаева

(Подпись)

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

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** составление программ с использованием массивов как формальных параметров подпрограмм.

**Теоретическая часть**

В *Object* *Pascal* можно в качестве параметров-значений, параметров-переменных и параметров-констант использовать массивы открытого типа, у которых не задаются размеры. В качестве фактического параметра в этом случае можно использовать массив любого размера, однако он должен состоять из компонент того же типа, который имеют и компоненты открытого массива.

Такие параметры введены для того, чтобы подпрограмма могла обрабатывать массив или строку любого размера. Фактический размер массива в этом случае может быть определен с помощью стандартной функции *High*. Открытый массив задается, как и обычный массив, но только без указания типа индекса. Следует иметь в виду, что индексация элементов открытого массива всегда начинается с нуля, а максимальный индекс элемента равен значению функции *High*. Фактический параметр может, однако, иметь и не такую индексацию компонент (индексы могут даже иметь другой порядковый тип), однако внутри подпрограммы индексы все равно приводятся к указанному выше виду. Внутри подпрограммы для задания значений открытому массиву можно использовать только его компоненты, массив целиком можно передавать в другие подпрограммы только как открытый массив или параметр без типа.

Форма описания:

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;

*Формат описания с использованием динамических массивов:*

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**

**Исходные данные**:

n=10

**Задание**

*i* изменяется от 1 до *n*. Составьте подпрограммы (по каждому пункту) для нахождения:

1. Максимального элемента последовательности и его индекса;
2. Количества элементов последовательности, превышающих по значению 1;
3. Суммы элементов этой последовательности с четными индексами;

*n* = 10.

**Программная реализация**

**program** lb8;

**const**

n = 10;

**type**

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

**var**

a: arr;

i: integer;

**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**

max := a[i];

result := i;

**end**;

**end**;

**function** p(b: arr): real;

**var**

i: integer;

**begin**

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

**if** a[i] > 1 **then**

result := result + 1;

**end**;

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

**var**

i: integer;

**begin**

**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**

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

write(a[i]:10:4);

**end**;

writeln;

writeln('Max[', max(a), '] = ', a[max(a)]:5:4);

writeln('Больше 1 = ', P(a));

writeln('Сумма четных = ', s(a):5:4);

**end**.

**Ответ**

2.1264 1.8736 0.9932 0.4617 0.8256 1.7795 2.4643 2.2623 1.3681 0.6103

Max[7] = 2.4643

Больше 1 = 6

Сумма четных = 6.9874

**Задание 2**

**Исходные данные**:

m=2.6e-4;

3.24, 11.2, 0.4, 6.23, 15.64

**Задание**

Составьте процедуру для определения наибольшего значения (PN) произведений элементов каждой строки матрицы В(5,3). Элементы матрицы В(5, 3) заполните случайными числами из интервала [1; 100].

В ответах приведите:

1. Значения элементов матрицы В(5, 3);
2. Значение PN;
3. Значение k.

**Программная реализация**

**program** lb8;

**const**

m = 0.26e-4;

**type**

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

**var**

b: arr;

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

i, j: integer;

PN, d, k: real;

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

**var**

i, j: integer;

mx: real;

**begin**

PN := 0;

mx := 1;

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

**begin**

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

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

**if** mx > PN **then**

**begin**

PN := mx;

**end**;

mx := 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]:6);

**end**;

writeln;

**end**;

f(b, PN);

writeln;

writeln('PN = ', PN);

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

d := d + ln(C[i]);

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

writeln('k = ', k);

**end**.

**Ответ**

9 28 76

62 43 56

83 45 16

43 34 24

50 14 99

PN = 149296

k = 7.31349471469104

**Задание 3**

**Исходные данные**:

*b = 0.294;*

*X[i] = 10, 20, 30, 40, 50;*

**Задание**

Составьте подпрограмму–функцию для вычисления суммы (SA) значений среднего арифметического элементов каждого столбца матрицы А(3,4). Элементы матрицы А(3, 4) заполните случайными числами из интервала [1; 9].

В ответах приведите:

1. Значения элементов матрицы A(5, 3);
2. Значение SA;
3. Значения элементов массива Z.

**Программная реализация**

**program** lb8;

**const**

b = 0.294;

**type**

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

**var**

A: arr;

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

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

i, j: integer;

s: real;

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

**var**

i, j: integer;

c: real;

**begin**

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

**begin**

c := 0;

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

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

result := result + c / 3;

**end**;

**end**;

**begin**

**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('SА = ', SA(A));

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

s := s + X[i] + b;

write('Z = [');

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

**begin**

Z[i] := (sqrt(X[i]) / b) + SA(A) \* sqrt(s);

write(Z[i]:7:1);

**end**;

write(']');

**end**.

**Ответ**

4 4 8 4

5 8 2 8

4 1 6 6

SА = 20

Z = [ 256.9 261.4 264.8 267.7 270.2]

**Выводы**

В ходе работы были составленные программы с использованием массивов как формальных параметров подпрограмм.