**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. 2Д91 Е.А. Резинкина

(Подпись)

13 апреля 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить теорию, выполнить задания, составив программы с использованием массивов как формальных параметров подпрограмм.

**Теоретическая часть**

Подпрограммы могут быть с параметрами и без параметров, т.е. список параметров может отсутствовать. В подпрограммах без параметров можно использовать как глобальные, так и локальные объекты.

Параметры подпрограмм позволяют при каждой активации (вызове) процедуры или функции работать с объектами (значениями, переменными, процедурами и функциями), заданными в точке активации, т.е. при вызове. В подпрограммах с параметрами в список параметров включают объекты вызываемой программы, которые при разных вызовах подпрограммы будут разными. Например, в списке параметров обязательно должно быть имя массива, если при вызове подпрограммы для обработки передается то массив А, то массив В. Если при всех вызовах внутренней подпрограммы имена обрабатываемых ею данных и имена результатов одинаковы, т.е. подпрограмма вызывается для одних и тех же фактических параметров, то целесообразно рассмотреть возможность использования подпрограммы без параметров, с использованием глобальных переменных.

Все формальные параметры должны быть определены в списке формальных параметров заголовка подпрограммы под своими именами, известными в блоке процедуры или функции, а также определен тип параметров.

При вызове подпрограммы фактические параметры должны быть в той же последовательности, что и соответствующие им формальные.

В Паскале различают следующие формальные параметры: параметры-значения, параметры-переменные, параметры-константы, бестиповые параметры, параметры-массивы, строки открытого типа, процедурные параметры.

Параметры-значения передаются основной программой в вызываемую подпрограмму через стек в виде их копий. Поэтому фактический параметр подпрограммой измениться не может.

При передаче в подпрограмму из основной программы параметров-переменных фактически через стек передаются их адреса в порядке, объявленном в заголовке подпрограммы. Поэтому подпрограмма имеет доступ к этим параметрам и может их изменить.

Следовательно, входные параметры подпрограммы могут быть как параметрами-значениями, так и параметрами-переменными. Выходные же - только параметрами-переменными.

Нецелесообразно использовать в качестве параметров-значений массивы больших размеров, так как это приведет к существенным дополнительным затратам ОП и времени на их копирование из ОП вызывающей программы в ОП стека для вызванной подпрограммы при многократных обращениях к этой подпрограмме.

**Практическая часть**

**Задание 1:**

Последовательность элементов задана общей формулой:

где *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** lb08\_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('Макс эл-т = ', b[max(b)], ' и его индекс = ', max(b));

writeln('Кол-во эл-тов, превышающих по значению 1 = ', posl(b));

writeln('Сумма эл-тов 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

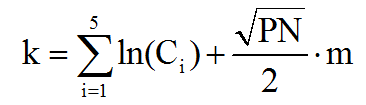
Макс эл-т = 2.46427125392093 и его индекс = 7

Кол-во эл-тов, превышающих по значению 1 = 6

Сумма эл-тов с четными индексами = 6.98742353540703

**Задание 2:**

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]. Полученное значение PN используйте при вычислении:

**

*m = 2,6e-4;*

*Ci = 3.42, 11.2, 0.4, 6.23, 15.64;*

В ответах приведите:

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

**Программная реализация**

**program** lb08\_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('PN = ', 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

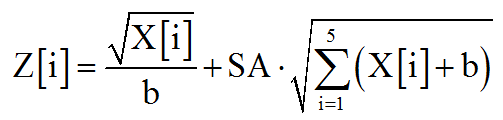
26 49 12

PN = 523320

k = 7.40251478318846

**Задание 3:**

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]. Значение SA используйте при вычислении:

*b = 0.294;*

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

В ответах приведите:

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

**Программная реализация**

**program** lb08\_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('Значение SA =', SA(a));

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

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

writeln;

writeln('Значения элементов массива 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

Значение SА = 17

Значения элементов массива Z:

219.98 224.44 227.85 230.74 233.28

**Выводы**

В ходе работы была изучена теория, а также составлены программы с использованием массивов как формальных параметров подпрограмм.