**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 А.А. Циттель

(Подпись)

\_\_07\_\_\_ \_\_апреля\_\_\_\_\_2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** Рассмотреть и составить программы с использованием массивов как формальных параметров подпрограмм.

**Теоретическая часть**

**Применение подпрограмм даёт следующие преимущества:**

1. Экономия памяти: каждая программная единица существует в основной программе в единственном экземпляре, в то время как обращаться к ней можно многократно из разных точек в одной или в разных программах;

2. Заключается в применении методики нисходящего проектирования программ. В этом случае алгоритм представляется в виде последовательности относительно крупных подпрограмм, реализующих самостоятельные смысловые части алгоритма. Подпрограммы в свою очередь разбиваются на менее крупные подпрограммы нижнего уровня и т. д. Последовательное структурирование программы продолжается до тех пор, пока реализуемые подпрограммами алгоритмы не станут настолько простыми, чтобы их можно было легко запрограммировать. Такие программные единицы легче тестировать и отлаживать, и у них более чёткая логическая структура.

Для организации структурной программы используют подпрограммы, которые делятся на **пользовательские** и **стандартные**, последние находятся в модуле turbo.tpl. Например, в модуле CRT находится процедура очистки экрана – CLRSCR; и процедура, отвечающая за цвет выводимого текста – TEXTCOLOR (). Стандартные процедуры – READ, WRITE и функции – SIN, ORD, CHR.

Каждая подпрограмма состоит из набора операторов, которые снабжены одним именем. По этому имени происходит **обращение** к данной подпрограмме.

По способу организации подпрограммы делятся на подпрограммы-процедуры – PROCEDURE и  подпрограммы – функции – FUNCTION (причём, функция – частный случай процедуры).

**Описать подпрограмму** – это означает, в разделе описаний основной программы после раздела объявления переменных (VAR) указать заголовок и тело программной единицы. В заголовке объявляются имя подпрограммы и формальные параметры, если они есть. Для функции указывается, кроме того, и тип возвращаемого ею результата. За заголовком следует тело подпрограммы, которое, подобно основной программе, состоит из раздела описаний и раздела исполняемых операторов. В разделе описаний подпрограммы могут встретиться описания подпрограмм низшего уровня, в тех – описания других подпрограмм и так далее. Все имена, описанные внутри программной единицы, локализуются в ней, то есть «невидимы» снаружи программы.

**Вызовом** подпрограммы называется упоминание имени этой подпрограммы в теле основной программы с указанием списка фактических параметров, если они есть. Это приводит к активизации программной единицы, и выполнению входящих в неё операторов. После выполнения последнего из них управление возвращается обратно в основную программу, и выполняются операторы, стоящие непосредственно за оператором вызова процедуры. Функция отличается от процедуры тем, что результат её работы возвращается в виде значения этой функции, и, следовательно, вызов функции может использоваться наряду с другими операндами в выражениях.

**Формат 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**;

**Пример:**

Составить процедуру для определения элементов массива В(5,3) как квадратов от элементов массива С(5,3). Определить среднее арифметическое элементов массива В(5,3). Элементы массива С(5,3) считать из файла, результаты расчета вывести в файл.

Исходные данные: 0.22 0.23 0.18

0.59 0.82 0.15

0.27 0.05 0.94

0.99 0.59 0.05

0.21 0.22 0.94

**type**

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

**var**

i, j: integer;

b, c: arr;

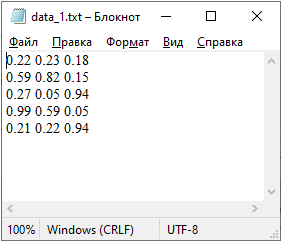
average: real;

f1, f2: text;

**procedure** get\_avarege(c: arr; **var** b: arr; **var** average: real);

**var**

i, j: integer;



**begin**

average := 0;

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

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

**begin**

b[i, j] := sqr(c[i, j]);

average := average + b[i, j]

**end**;

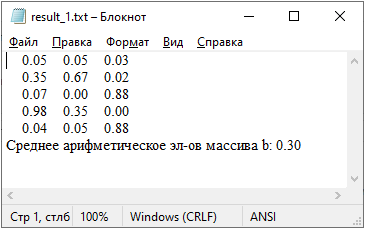
average := average / (i \* j)

**end**;

**begin**

assign(f1, 'data\_1.txt');

reset(f1);

 assign(f2, 'result\_1.txt');

rewrite(f2);

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

**begin**

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

read(f1, c[i, j]);

readln(f1)

**end**;

get\_average(c, b, average);

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

**begin**

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

write(f2, b[i, j]:4:2);

writeln(f2)

**end**;

writeln(f2, 'Среднее арифметическое эл-тов массива b: ', average:4:2);

close(f1);

close(f2)

**end**.

**Практическая часть**

**Задание 1**

**Исходные данные**: Последовательность элементов задана общей формулой:

*a[i] = arctg (2 \* i + i / n) – sin (i + n)*, где *i* изменяется от 1 до *n*.

**Задание**

Составьте подпрограммы (по каждому пункту) для нахождения:

* Максимального элемента последовательности и его индекса;
* Количества элементов последовательности, превышающих по значению 1;
* Суммы элементов этой последовательности с четными индексами;
* *n* = 10.

**Программная реализация**

**Program** lab8;

**var**

i, imax, n: integer;

A: **array of** real;

f: text;

max: real;

**procedure** max\_element(A: **array of** real; n: integer; **var** imax: integer; **var** max: real);

**var**

i: integer;

**begin**

max := A[0];

imax := 0;

**for** i := 1 **to** 9 **do**

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

**begin**

max := A[i];

imax := i

**end**

**end**;

**function** more\_than\_one(A: **array of** real): integer;

**var**

i: integer;

**begin**

result := 0;

**for** i := 0 **to** 9 **do**

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

result := result + 1;

**end**;

**function** sum\_ch(A: **array of** real): real;

**var**

i: integer;

**begin**

result := 0;

**for** i := 0 **to** 9 **do**

**if** i **mod** 2 = 0 **then**

result := result + A[i];

**end**;

**begin**

n := 10;

assign(f, 'task1output.txt');

rewrite(f);

SetLength(A, n);

**for** i := 0 **to** 9 **do**

**begin**

A[i] := arctan(2 \* i + i / n) - sin(1 + n);

writeln(f, A[i]:4:2)

**end**;

max\_element(A, n, imax, max);

writeln(f, '1.');

writeln(f, 'Максимальный член последовательности: ', max);

writeln(f, 'Его порядковый номер равен: ', imax);

writeln(f, '2.');

writeln(f, 'Количество членов последовательности, больших единицы: ', more\_than\_one(A));

writeln(f, '3.');

writeln(f, 'Сумма членов последовательности с четными индексами: ', sum\_ch(A));

close(f);

**end**.

**Ответ:**

1.00

2.13

2.34

2.41

2.45

2.48

2.49

2.50

2.51

2.52

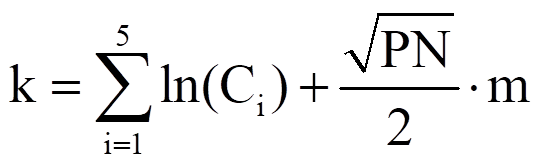
Максимальный член последовательности: 2.51792577110309

Его порядковый номер равен: 9

Количество членов последовательности, больших единицы: 9

Сумма членов последовательности с четными индексами: 10.792250469628

**Задание 2**

**Исходные данные**:

*m = 2,6e-4;*

*Ci = 3.42, 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].
* Полученное значение PN используйте при вычислении: (исходные данные).

**В ответах приведите**

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

**Программная реализация**

**Program** lab8;

**type**

arr = **array** [0..2, 0..4] **of** integer;

**var**

i, j: integer;

B: arr;

f: text;

k, m, sum: real;

**const**

C: **array** [0..4] **of** real = (3.42, 11.2, 0.4, 6.23, 15.64);

**function** PN\_val(B: arr): longint;

**var**

i, j: integer;

mult: integer;

C: **array** [0..2] **of** integer;

**begin**

mult := 1;

**for** i := 0 **to** 2 **do**

**begin**

**for** j := 0 **to** 4 **do**

mult := mult \* B[i, j];

C[i] := mult;

mult := 1;

**end**;

result := C[0];

**for** i := 1 **to** 2 **do**

**if** C[i] > result **then**

result := C[i]

**end**;

**begin**

assign(f, 'task2output.txt');

rewrite(f);

writeln(f, 'Массив B:');

m := 0.294;

sum := 0;

**for** i := 0 **to** 2 **do**

**begin**

**for** j := 0 **to** 4 **do**

**begin**

B[i, j] := random(99) + 1;

write(f, B[i, j]:4)

**end**;

writeln(f)

**end**;

**for** i := 0 **to** 4 **do**

sum := sum + C[i];

k := sum + sqrt(PN\_val(B) / 2 \* m);

writeln(f, 'Значение PN равно ', PN\_val(B));

writeln(f, 'Значение k равно ', k);

close(f);

**end**.

**Ответ:**

Массив B:

11 79 39 58 95

50 54 96 10 6

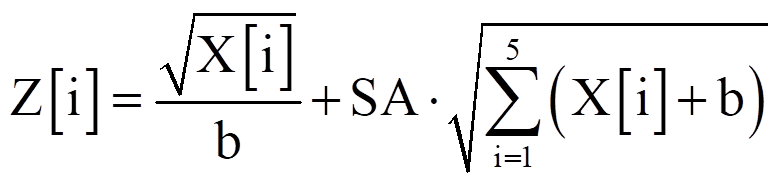
5 15 28 59 13

Значение PN равно 186739410

Значение k равно 5276.2309194287

**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];
* Значение SA используйте при вычислении: (исходные данные).

**В ответах приведите**

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

**Программная реализация**

**Program** lab8;

**type**

arr = **array** [0..2, 0..3] **of** integer;

**var**

i, j: integer;

A: arr;

Z: **array** [0..4] **of** real;

f: text;

b, sum: real;

**const**

X: **array** [0..4] **of** integer = (10, 20, 30, 40, 50);

**function** SA\_val(A: arr): real;

**var**

i, j: integer;

sred, sum: integer;

**begin**

sred := 0;

result := 0;

**for** j := 0 **to** 3 **do**

**begin**

**for** i := 0 **to** 2 **do**

sred := sred + A[i, j];

result := result + sred / 3;

sred := 0

**end**;

**end**;

**begin**

assign(f, 'task3output.txt');

rewrite(f);

writeln(f, 'Массив А:');

b := 0.294;

sum := 0;

**for** i := 0 **to** 2 **do**

**begin**

**for** j := 0 **to** 3 **do**

**begin**

A[i, j] := random(8) + 1;

write(f, A[i, j]:2)

**end**;

writeln(f)

**end**;

**for** i := 0 **to** 4 **do**

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

**for** i := 0 **to** 4 **do**

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

writeln(f, 'Значение SA равно ', SA\_val(A));

writeln(f, 'Массив Z:');

**for** i := 0 **to** 4 **do**

write(f, Z[i]:8:2);

close(f);

**end**.

**Ответ:**

Массив А:

3 3 5 1

8 1 8 5

4 1 3 1

Значение SA равно 14.3333333333333

Массив Z:

187.16 191.62 195.03 197.92 200.46

**Выводы**

В ходе работы мы рассмотрели и составили программы с использованием массивов как формальных параметров подпрограмм.