**Лабораторная работа 12.**

**1) Тема лабораторной работы:**

Многоступенчатые циклические вычислительные процессы. Двумерные массивы.

**2) Цель:**

Научиться работать с многоступенчатыми циклическими вычислительными процессами и двумерными массивами средствами языка Pascal и компилятора Pascal ABC.

**3) Используемое оборудование:**

Компьютер, компилятор Pascal ABC, язык Pascal.

**Задача 1**

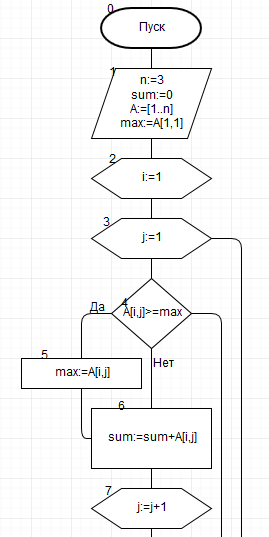
**4) Постановка задачи:**

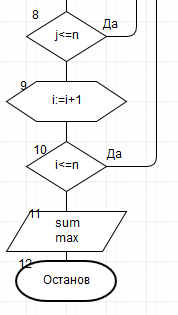
Найти сумму всех элементов массива 3x3. Массив задается явно внутри программы. Найти максимальный элемент.

**5) Математическая модель:**

Примем что первый элемент массива максимальный и будем сравнивать его со следующими элементами массива, если следующий элемент окажется больше максимального, перезапишем его в переменную максимума. Проходя по массиву будем складывать его элементы и получим сумму всех его элементов.

**6) Блок-схема:**





**7) Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| N | Кол-во строк и столбцов массива | Integer |
| Sum | Сумма всех элементов массива | Integer |
| Max | Максимальный элемент массива | Integer |
| I,j | Счетчики внешнего и внутреннего циклов | Integer |
| A | Двумерный массив | integer |

**8) Код программы**

**program** lr12;

**const**

n = 3;

**var**

A: **array** [1..n, 1..n] **of** integer;

i, j, sum, max: integer;

**begin**

sum := 0;

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

**begin**

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

**begin**

A[i, j] := random(100);

**end**;

**end**;

max := A[1,1];

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

writeln(A[i]);

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

**begin**

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

**begin**

**if** A[i, j] >= max **then**

max := A[i, j];

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

**end**;

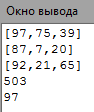
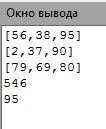
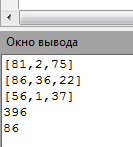
**end**;

writeln(sum);

writeln(max);

**end**.

**9) Результат работы программы:**

**Задача 2**

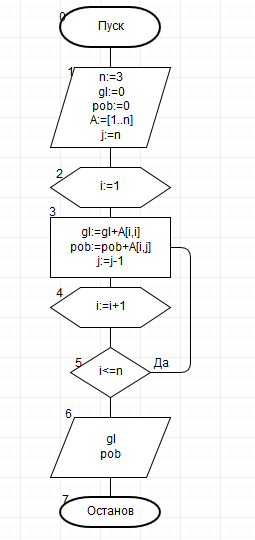
**4) Постановка задачи:**

Дан массив 3x3. Найти сумму элементов на главной диагонали и сумму элементов побочной диагонали.

**5) Математическая модель:**

Главная диагональ имеет формулу элемента вида a[i,i], а побочная диагональ вида a[i,j], при этом j идет от количества строк/столбцов до 1, а i от 1 до количества строк/столбцов.

**6) Блок-схема:**



**7) Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| N | Кол-во строк и столбцов массива | Integer |
| Gl, pob | Сумма элементов главной/побочной диагонали | Integer |
| I,j | Счетчик цикла, индекс элемента | Integer |
| A | Массив | integer |

**8) Код программы:**

**program** lr12;

**const**

n=3;

**var**

A: **array** [1..n, 1..n] **of** integer;

i, j, gl, pob: integer;

**begin**

gl := 0;

pob := 0;

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

**begin**

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

**begin**

A[i, j] := random(100);

**end**;

**end**;

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

writeln(A[i]);

j := n;

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

**begin**

gl := gl + A[i, i];

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

j := j - 1;

**end**;

writeln(gl);

writeln(pob);

**end**.

**program** lr12;

**const**

n=3;

**var**

A: **array** [1..n, 1..n] **of** integer;

i, j, gl, pob: integer;

**begin**

gl := 0;

pob := 0;

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

**begin**

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

**begin**

A[i, j] := random(100);

**end**;

**end**;

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

writeln(A[i]);

j := n;

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

**begin**

gl := gl + A[i, i];

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

j := j - 1;

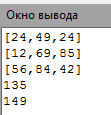
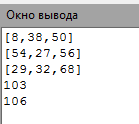
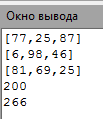
**end**;

writeln(gl);

writeln(pob);

**end**.

**9) Результат работы программы:**

**Задача 3**

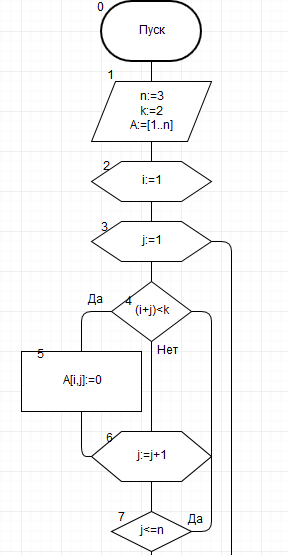
**4) Постановка задачи:**

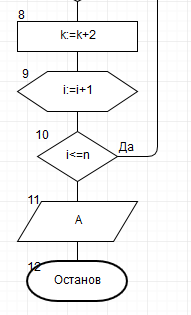
Дан массив 3x3. Заменить элементы, стоящие ниже главной диагонали нулями.

**5) Математическая модель:**

Первый элемент главной диагонали имеет сумму индексов 2, последующие элементы находятся по формуле .

**6) Блок-схема:**





**7) Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| A | Массив | Integer |
| N | Кол-во строк и столбцов массива | Integer |
| I,j | Счетчик внешнего и внутреннего цикла | Integer |
| K | Переменная проверки индекса на соответствие условию | Integer |

**8) Код программы:**

**program** lr12;

**const**

n=3;

**var**

A: **array** [1..n, 1..n] **of** integer;

i, j, k: integer;

**begin**

k := 2;

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

**begin**

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

**begin**

A[i, j] := random(100);

**end**;

**end**;

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

writeln(A[i]);

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

**begin**

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

**begin**

**if** (i + j )< k **then**

A[i, j] := 0;

**end**;

k := k + 2;

**end**;

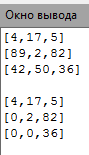
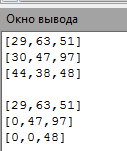
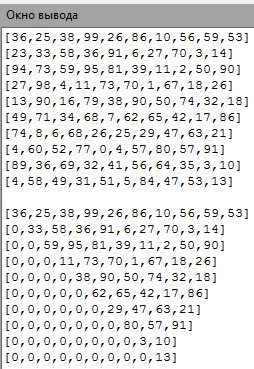
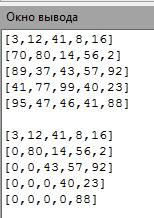
writeln();

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

writeln(A[i])

**end**.

**9) Результат работы программы:**

**Задача 4**

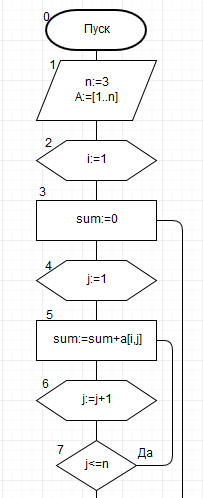
**4) Постановка задачи:**

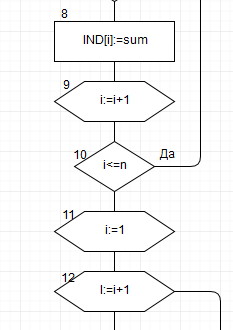
Дана матрица 3x3. Найти суммы элементов каждой строки и упорядочить строки по возрастанию согласно их суммам.

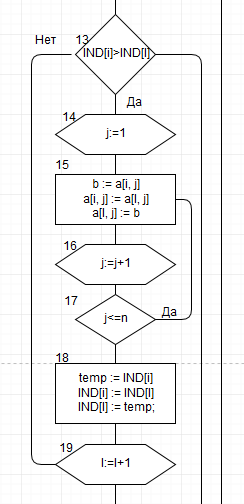
**5) Математическая модель:**

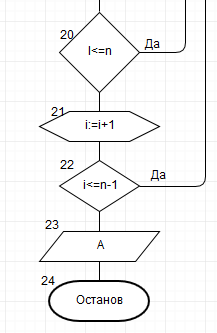
Посчитаем сумму чисел для каждой строки и занесем данные в массив, затем отсортируем строки по порядку возрастания элементов.

**6) Блок-схема:**









**7) Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| N | Кол-во строк и столбцов массива | Integer |
| A  IND | Исходный массив  Массив с суммой строки | Integer  Integer |
| I,j | Счетчики внешнего и внутреннего цикла | Integer |
| B | Временная переменная | integer |
| L | Счетчик внешнего цикла(2) | Integer |
| Sum | Сумма строки | Integer |
| Temp | Временная переменная для суммы | integer |

**8) Код программы:**

**program** lr12;

**const**

n = 3;

**var**

A: **array** [1..n, 1..n] **of** integer;

IND: **array** [1..n] **of** integer;

b, i, j, l, sum, temp: integer;

**begin**

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

**begin**

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

**begin**

A[i, j] := random(100);

**end**;

**end**;

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

writeln(A[i]);

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

**begin**

sum := 0;

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

**begin**

sum := sum + a[i, j];

**end**;

IND[i] := sum;

writeln(IND[i]);

**end**;

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

**begin**

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

**begin**

**if** IND[i] > IND[l] **then**

**begin**

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

**begin**

b := a[i, j];

a[i, j] := a[l, j];

a[l, j] := b;

**end**;

temp := IND[i];

IND[i] := IND[l];

IND[l] := temp;

**end**;

**end**;

**end**;

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

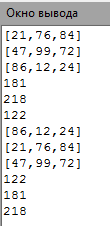
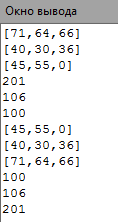
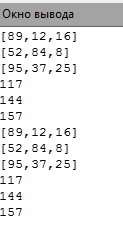
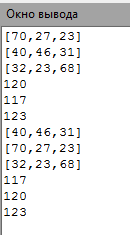
writeln(A[i]);

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

writeln(IND[i]);

**end**.

**9) Результат работы программы:**

**10) Вывод:**

В ходе лабораторной работы я научился работать с многоступенчатыми циклическими вычислительными процессами и двумерными массивами средствами языка Pascal и компилятора Pascal ABC.