**Условие задачи**

Дана матрица, состоящая из n строк и n столбцов или одномерные массивы из n элементов каждый. Составить алгоритм и программу для получения и вывода указанных в условии результатов и самих массивов, если изменялись какие-либо их элементы. Алгоритм любой задачи может быть составлен с использованием единственного двукратного цикла.

Получить массив Х1, Х2, ..., Хn. по правилу: Хi = 1, если элемент Аii больше каждого из элементов 1-й строки матрицы, иначе Хi = 0. Найти также сумму элементов матрицы.

**Решении задачи**

Прежде всего необходимо учесть, чтобы наш алгоритм соответствовал нескольким важным факторам:

* Использование алгоритма было удобным;
* Алгоритм был простым и наглядным;

**Метод решения** задачи состоит в получение массива Х1, Х2, ..., Хn. по правилу: Хi = 1, если элемент Аii больше каждого из элементов 1-й строки матрицы, иначе Хi = 0, а также нахождение суммы элементов матрицы.

**Состав данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| **Входные/Исходные данные** | | | |
| *matrix1* | матрица | двумерный динамический массив целочисленного типа | двумерный динамический массив целочисленного типа |
| *array1* | выходной массив | одномерный динамический массив целочисленного типа | одномерный динамический массив целочисленного типа |
| *n* | количество строк/столбцов | целочисленный тип данных | переменная |
| **Выходные данные** | | | |
| *array1[i]* | элемент, найденный по правилу из условия | целочисленный тип данных | переменная |
| *sum* | сумма элементов матрицы | целочисленный тип данных | переменная |
| **Промежуточные данные** | | | |
| *x, y* | временные переменная, выделенная под хранения индекса или переиспользуется под переменную цикла | целочисленный тип данных | переменная |
| *temp* | флаг | булевый | переменная |

**Форма ввода**

n

<*n*>

**Форма вывода**

<array[0]>..<array[i]>

sum <sum>

**Листинг программы, написанной на языке object pascal и скомпилированной с помощью freepascal**

**program** lab\_2\_5\_2;

**uses** Math, sysutils;

**type**

matrix = **array of array of** integer;

**var**

n, sum, x, y: integer;

temp: boolean;

matrix1: matrix;

array1: **array of** integer;

**begin**

writeln('n');

read(n);

setLength(matrix1, n, n);

setLength(array1, n);

n := n - 1;

randomize;

**for** x := 0 **to** n **do**

**begin**

array1[x] := 0;

**for** y := 0 **to** n **do**

**begin**

matrix1[x, y] := random(10);

**end**;

**end**;

**for** x := 0 **to** n **do**

**begin**

temp := true;

**for** y := 0 **to** n **do**

**begin**

**if** matrix1[x, x] < matrix1[x, y] **then begin**

temp := false;

**end**;

**end**;

sum := sum + matrix1[x, y];

**end**;

**if** temp = true **then begin**

array1[x] := 1;

write(array1[x], ' ');

**end**;

**end**.