*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

|  |  |  |  |
| --- | --- | --- | --- |
| Gerb-BMSTU_01 | **«Московский государственный технический университет имени Н.Э. Баумана»** | |  |
| Факультет: | | ИУ «Информатика и системы управления» |
| Кафедра: | | ИУ-6 «Компьютерные системы и сети» |

**Отчет**

**по домашнему заданию № 6-Б**

**«Матрицы»**

вариант №4

**Дисциплина: Основы программирования**

|  |  |  |
| --- | --- | --- |
| Студент группы ИУ6-51 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Милков В. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гукетлев Д. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |

*Москва, 2016*

**Содержание**

Задание 3

Блок-схема алгоритма 4

[Код программы 5](#h.30j0zll)

[Скриншоты выполнения программы 6](#h.1fob9te)

[Вывод 7](#h.2et92p0)

**Задание**

Дана целочисленная матрица R(n,n), n<11 и целые числа p и q (1 p 9; 1 q 9). Определить сколько элементов, равных элементу rpq, содержит матрица и удалить строку с этим элементом.

**Блок-схема алгоритма**



**Код программы**

**program main;**

**const**

**DIMENSION = 20;**

**var**

**i, j, p, q, seeking, total, lines: integer;**

**isFound: boolean;**

**matrix: array of array[1..DIMENSION] of integer;**

**begin**

**total := 0;**

**lines := 0;**

**setlength(matrix, DIMENSION);**

**for i:=1 to length(matrix) - 1 do begin**

**for j:=1 to length(matrix[i]) do begin**

**matrix[i][j] := random(20);**

**end;**

**end;**

**repeat write('Input q (1 ≤ q ≤ 9): '); read(q); until (q >= 1) and (q <= 9);**

**repeat write('Input p (1 ≤ p ≤ 9): '); read(p); until (p >= 1) and (p <= 9);**

**writeln;**

**seeking := matrix[p][q];**

**writeln('Seeking matrixs element is: ', seeking);**

**for i:=1 to length(matrix) - 1 do begin**

**writeln;**

**if lines > 0 then begin**

**matrix[i - lines] := matrix[i];**

**end;**

**isFound := false;**

**for j:=1 to length(matrix[i]) do begin**

**write(matrix[i][j], chr(9));**

**if matrix[i][j] = seeking then begin**

**isFound := true;**

**inc(total);**

**end;**

**end;**

**if isFound then begin**

**inc(lines);**

**write(' ← Element is found');**

**end;**

**end;**

**setlength(matrix, length(matrix) - lines);**

**writeln;**

**writeln;**

**for i:=1 to length(matrix) - 1 do begin**

**for j:=1 to length(matrix[i]) do begin**

**write(matrix[i][j], chr(9));**

**end;**

**writeln;**

**end;**

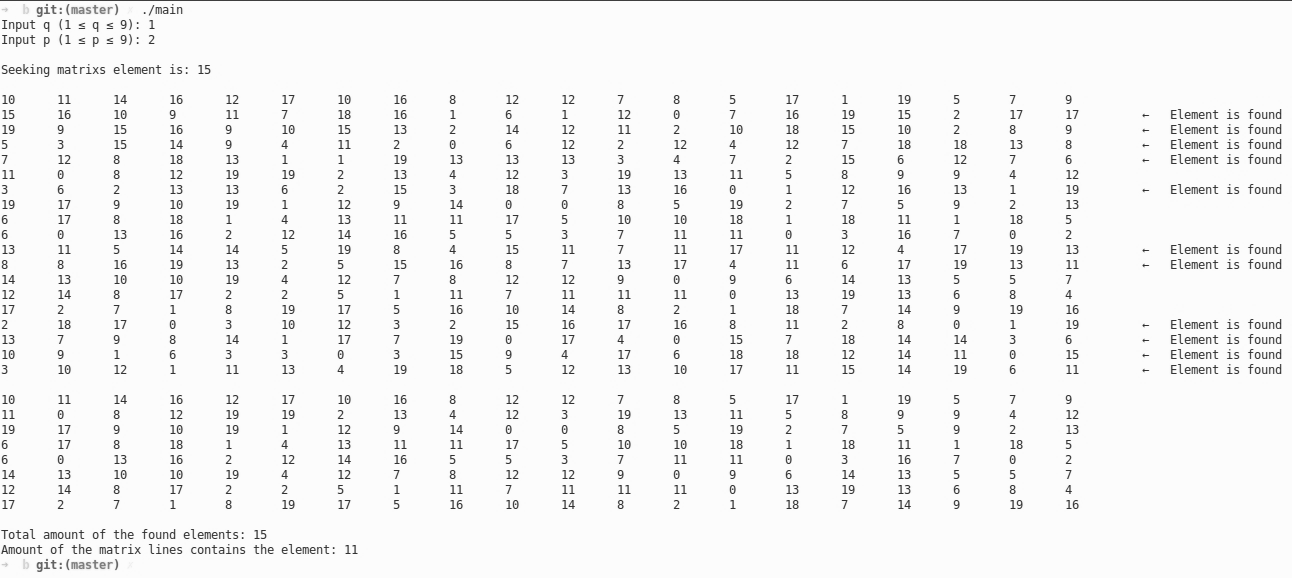
**writeln;**

**writeln('Total amount of the found elements: ', total);**

**writeln('Amount of the matrix lines contains the element: ', lines);**

**end.**

**Скриншоты выполнения программы**



**Вывод**

Двумерный массив в Паскале трактуется как одномерный массив, тип элементов которого также является массивом (массив массивов). Положение элементов в двумерных массивах Паскаля описывается двумя индексами. Их можно представить в виде прямоугольной таблицы или матрицы.

Для последовательного ввода элементов одномерного массива мы использовали цикл for, в котором изменяли значение индекса с 1-го до последнего. Но положение элемента в двумерном массиве Паскаля определяется двумя индексами: номером строки и номером столбца. Это значит, что нам нужно будет последовательно изменять номер строки с 1-й до последней и в каждой строке перебирать элементы столбцов с 1-го до последнего. Значит, нам потребуется два цикла for , причем один из них будет вложен в другой.

Двумерный массив Паскаля можно заполнить случайным образом, т.е. использовать функцию random (N), а также присвоить каждому элементу матрицы значение некоторого выражения. Способ заполнения двумерного массива Паскаля выбирается в зависимости от поставленной задачи, но в любом случае должен быть определен каждый элемент в каждой строке и каждом столбце.

В данной задаче для экономии памяти мы не создаем новый массив, а модифицируем существующий.