

```

1      {
      MSP, 10.05.07, 15:48

      Эта программа работает с модулем work_dm.pas

      Использование адресной арифметики при работе с двумерными
      массивами, размещаемыми в динамической памяти.

      1. Задана матрица F (6, 7). Найти в каждой строке матрицы
      максимальный и минимальный элементы и поместить их на место
      первого и последнего элемента строки соответственно.
      Диапазон [-50; 50]. Использовать модуль work_dm (см. выше).
      }

11     Program matrix_1;

12     Uses
13         Crt,
14         work_dm;

15     Const
16         N = 6; { число строк }
17         M = 7; { число столбцов }

18     Type
19         { массив ВСЕГО для ОДНОГО элемента }
20         mas = array [1..1] of Integer;

21     Var
22         p { указатель на начало массива }
23         : ^mas;

24         max, { максимальный элемент строки матрицы }
25         min { минимальный элемент строки матрицы }
26         : Integer;

27         i, { индекс строк }
28         j { индекс столбцов }
29         : word;

30         S
31         : longint;

32     Begin
33         ClrScr;

34         {
            Создадим в динамической памяти массив, содержащий
            M вещественных чисел.
        }

```

```

38     Randomize;

39     {
        директива компилятору о том, что не нужно ему обращать
        внимание
        на то, что произойдет выход за пределы массива
    }
44     {$R-}

45     {
        I. Создадим динамический двумерный массив N x M
    }

48     { у двумерного массива N строк }
49     WriteLn ('Исходная матрица:');
50     S := 0;
51     for i := 1 to N do
52     begin
53         {
            У двумерного массива M столбцов,
            выделим память под одномерный массив
        }
57         GetMem (p, M * SizeOf (Integer));

58         P_str [i] := p;

59         S := S + M * SizeOf (Integer);

60         {
            заполним массив случайным образом
        }
63         for j := 1 to M do
64             p^ [j] := Random (101) - 50;

65         { печать i-той строки массива }
66         for j := 1 to M do
67             Write (p^ [j]:4);
68             WriteLn;
69         end;

70     WriteLn;
71     WriteLn ('Под массив OS выделила ', S, ' bytes памяти. ');
72     ReadLn;

73     {
        II. Найдем в каждой строке матрицы максимальный
        и минимальный элемент.
        В 1 элемент строки запишем максисмальный,
        а в последний - минимальный.
    }

```

```

    }

79     for i := 1 to N do
80         begin
81             max := get_elem (i, 1);
82             min := get_elem (i, 1);

83             for j := 1 to M do
84                 begin
85                     if get_elem (i, j) > max
86                         then
87                             max := get_elem (i, j);

88                     if get_elem (i, j) < min
89                         then
90                             min := get_elem (i, j);
91                 end;

92             put_elem (i, 1, max);
93             put_elem (i, M, min);
94         end;

95     {
        печать результата
    }
96     WriteLn;
97     WriteLn ('Матрица-результат:');
98     for i := 1 to N do
99         begin
100             for j := 1 to M do
101                 Write (get_elem (i, j):4);
102             WriteLn;
103         end;

104     {
        Вспомним ПРАВИЛО о том, что в конце работы каждой программы
        нужно возвращать использованную память назад OS...

        Вернем память OS, занятую динамическим массивом.
        Уничтожим одномерный динамический массив.
    }
105     S := 0;
106     for i := 1 to N do
107         begin
108             FreeMem (P_str [i], M * SizeOf (Integer));
109             S := S + M * SizeOf (Integer);
110         end;

111     WriteLn;
112     WriteLn;
113     WriteLn ('OS вернули: ', S, ' bytes памяти.');
```

```
121     ReadLn;  
122     End.
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-1-task-590.html>

Sergey Mitrofanov, 28.08.14, 20:34

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014