

### Вариант 3

Пользователь вводит целые числа  $n$ ,  $m$ ,  $max$ ,  $min$ :

- $n$  — количество строк в матрице
- $m$  — количество столбцов в матрице
- $min$  — минимальное значение элемента матрицы
- $max$  — максимальное значение элемента матрицы

Необходимо сформировать массив массивов (назовем его  $A$ ), содержащий  $m$  одномерных целочисленных массивов со случайной длиной от 1 до  $n$ . Создать другой массив массивов (назовем его  $B$ ), тоже содержащий  $m$  одномерных целочисленных массивов с такой же длиной, какую имеет соответствующий массив из  $A$ . Каждый одномерный массив (подмассив) в  $A$  и  $B$  должен быть заполнен случайными вещественными числами от  $min$  до  $max$ . По этим двум массивам массивов построить другой массив массивов  $minAB$ , заполненный минимальными значениями среди элементов, стоящих на соответствующих позициях в массивах массивов  $A$  и  $B$ . К примеру, если  $A[1][5] = 6$ , а  $B[1][5] = 3$ , то  $minAB[1][5] = 3$ . Вывести все 3 массива массивов.

Для этого реализуйте и используйте следующие методы:

- **void PrintJaggedArray(double[][] arr)** — метод для печати массива массивов. Элементы исходного массива печатать на новой строке, а элементы подмассивов через пробел (см. пример ниже).
- **double[][] CreateJaggedArrayOfMins(double[][] a, double[][] b)** — находит массив массивов  $minAB$ .
- **double[][] CreateArray(int n, int m, int min, int max)** - создает массив массивов длины  $m$ . Каждый элемент массива массивов имеет случайную длину от 1 до  $n$  (включительно), и заполняется случайными целыми числами от  $min$  до  $max$ ;

#### Проверьте себя:

При вводе  $n = 5, m = 5, min = 10, max = 200$  может быть следующий вывод:

A

```
89,330 153,600 11,596 67,344 77,211
36,977 188,273
149,699 95,558 143,084 179,993
195,181
14,205
```

B

```
135,878 166,039 171,729 122,568 18,673
70,070 50,524
165,005 89,611 137,980 156,884
13,892
120,005
```

minAB

```
89,330 153,600 11,596 67,344 18,673
36,977 50,524
149,699 89,611 137,980 156,884
13,892
14,205
```