

# Яндекс.Практикум, Алгоритмы: Спринт 1.

## Базовые структуры данных

🕒 1 авг 2020, 19:08:53  
старт: 1 авг 2020, 18:02:26

### В. Мониторинг

	Все языки	Oracle Java 8	Node JS 8.16
Ограничение времени	1 секунда	8 секунд	11.5 секунд
Ограничение памяти	64Mb	95Mb	362Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

Алла получила задание, связанное с мониторингом работы различных серверов. Требуется понять, сколько времени обрабатываются определённые запросы на конкретных серверах. Эту информацию нужно хранить в матрице, где номер столбца соответствуют идентификатору запроса, а номер строки — идентификатору сервера. Алла перепутала строки и столбцы местами. С каждым бывает. Помогите ей исправить баг.

Есть матрица размера  $m \times n$ . Нужно написать функцию, которая её транспонирует.

Транспонированная матрица получается из исходной замены строк на столбцы.

Например, для матрицы A

```
[[1, 2, 3]
 [0, 2, 6]
 [7, 4, 1]
 [2, 7, 0]]
```

транспонированной будет следующая матрица:

```
[[1, 0, 7, 2]
 [2, 2, 4, 7]
 [3, 6, 1, 0]]
```

### Формат ввода

В первой строке задано число  $n$  - количество строк матрицы. Во второй строке -  $m$  - число столбцов,  $m$  и  $n$  не превосходит 1000. В следующих  $n$  строках задана матрица. Числа в матрице по модулю не превосходят 1000.

### Формат вывода

Напечатайте транспонированную матрицу в том же формате.

#### Пример 1

Ввод	Вывод
4	1 0 7 2
3	2 2 4 7
1 2 3	3 6 1 0
0 2 6	
7 4 1	
2 7 0	

#### Пример 2

## Ввод

```
9
5
-7 -1 0 -4 -9
5 -1 2 2 9
3 1 -8 -1 -7
9 0 8 -8 -1
2 4 5 2 8
-7 10 0 -4 -8
-3 10 -7 10 3
1 6 -7 -5 9
-1 9 9 1 9
```

## Вывод

```
-7 5 3 9 2 -7 -3 1 -1
-1 -1 1 0 4 10 10 6 9
0 2 -8 8 5 0 -7 -7 9
-4 2 -1 -8 2 -4 10 -5 1
-9 9 -7 -1 8 -8 3 9 9
```

Язык GNU c++17 7.3

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4 #include <sstream>
5
6 using namespace std;
7
8 typedef unsigned short UShort;
9
10 int main()
11 {
12     string line;
13     getline( cin, line );
14     UShort rows = static_cast<UShort>( stoi( line ) );
15
16     getline( cin, line );
17     UShort cols = static_cast<UShort>( stoi( line ) );
18
19     short** matrix = new short*[ rows ];
20     for( UShort i = 0; i < rows; ++i )
21     {
22         getline( cin, line );
23         stringstream sstream( line );
24         matrix[ i ] = new short[ cols ];
25         for( short elem, ind = 0; sstream >> elem; ++ind )
26             matrix[ i ][ ind ] = elem;
27     }
28
29     short **transposed = new short*[ cols ];
30     for( short i = 0; i < cols; ++i )
31         transposed[ i ] = new short[ rows ];
32
33     for( UShort i = 0; i < rows; ++i )
34         for( UShort j = 0; j < cols; ++j )
35             transposed[j][i] = matrix[i][j];
36
37     for( UShort i = 0; i < cols; ++i )
38     {
```

Отправить

Предыдущая

Следующая