

1)

Напишите функцию, которая принимает на вход целочисленную матрицу M (другими словами, просто двумерный целочисленный массив) размера $rows \times cols$, и возвращает транспонированную матрицу MT (тоже двумерный целочисленный массив) размера $cols \times rows$. Если в M на пересечении i -ой строки и j -ого столбца стояло число X , то на пересечении j -ой строки и i -ого столбца в матрице MT тоже будет стоять число X , или другими словами $MT[j][i] = M[i][j]$

.

Обратите внимание, что вам неизвестно, каким именно способом выделялась память для массива M

. Выделять память под массив MT

можете любым удобным вам способом. Изменять исходную матрицу нельзя.

Требования к реализации: при выполнении этого задания вы можете определять любые вспомогательные функции. Вводить или выводить что-либо не нужно. Реализовывать функцию `main` не нужно.

```
int ** transpose(const int * const * m, unsigned rows, unsigned cols)
{
    /* ... */
}
```

2)

Реализуйте функцию `swap_min`, которая принимает на вход двумерный массив целых чисел, ищет в этом массиве строку, содержащую наименьшее среди всех элементов массива значение, и меняет эту строку местами с первой строкой массива.

Подумайте, как обменять строки массива, не обменивая элементы строк по-отдельности.

Требования к реализации: при выполнении этого задания вы можете определять любые вспомогательные функции. Вводить или выводить что-либо не нужно. Реализовывать функцию `main` не нужно.

```
void swap_min(int *m[], unsigned rows, unsigned cols)
{
    /* ... */
}
```