



### Lista de exercícios – Ponteiros 03 - Matrizes

Implemente a função descrita nas questões a seguir, seguindo as especificações. No mesmo exercício, construa a função main() que utilize ou teste a função desenvolvida.

1. void m\_escala(int \*mat, int k, int lin, int col);  
Função : Multiplica a matriz mat pelo escalar k. Armazena na própria matriz  
Parametros : mat – ponteiro para matriz origem  
                  k – valor escalar multiplicando de mat  
                  lin,col – numero de linhas e colunas da matriz mat
2. void m\_identidade(int \*mat, int ordem);  
Função : preenche a matriz mat como matriz identidade (diagonal principal =1 e demais =0)  
Parametros : mat – ponteiro para a área de armazenamento da matriz quadrada.  
                  Ordem – tamanho da matriz (num linhas = num colunas)
3. int m\_transposta (int \*mat\_t, int \*mat,int lin, int col);  
Função : Gera a matriz transposta da matriz de entrada mat  
Parametros : mat\_t – ponteiro para a área da matriz transposta  
                  mat – ponteiro para area da matriz de entrada/origem  
                  lin,col – numero de linhas e colunas da matriz origen  
Retorno : quantidade de valores copiados para p. 0 se ocorreu erro.  
DICA : declare variável para armazenar os dados apontados por mat\_t no programa principal.
4. void m\_ordena\_linha(int \*dados, int lin, int col);  
Função : Ordena em ordem crescente os valores de cada linha da matriz dados  
                  Cada linha deve ser ordenada independente das demais.  
Parametros : dados – ponteiro para vetor a ser ordenado  
                  lin, col – numero de linhas e colunas de dados  
DICA : Utilize um algoritmo de ordenação conhecido, como o bubble sort ou o quick-sort