

TAREFA – MATRIZ 2D ALOCADA DINAMICAMENTE

Considere matrizes 2D (quadradas ou não) alocadas dinamicamente e contendo números inteiros, como o exemplo a seguir que apresenta uma matriz quadrada diagonal¹:

$$v = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$


Primeiro defina as constantes simbólicas TRUE e FALSE no seu código. Depois, escreva e teste as seguintes funções:

- Função que, dados os números de linhas e colunas, retorna uma matriz 2D de inteiros (quadrada ou não) com zeros (alocada dinamicamente).
- Função para imprimir uma matriz 2D linha a linha (quadrada ou não);
- Função que recebe uma matriz 2D de zeros e uma constante inteira e altera a matriz tornando-a diagonal com essa constante (e sem criar novos espaços de memória). Se a matriz não é uma matriz quadrada, a função retorna a constante simbólica FALSE;
- Função que verifica se uma matriz dada é diagonal². A função também retorna FALSE se a matriz recebida não é uma matriz quadrada.
- Função que libera toda a memória alocada.

Na *main*, quando testar a função que verifica se a matriz é diagonal, use obrigatoriamente o operador ternário (*var = BooleanExpression ? Expression1 : Expression2;*) para imprimir o resultado como **S** se for diagonal e **N** caso contrário.

Veja, a seguir, um exemplo de saída (porém teste com dimensão e valores diferentes):

```
Microsoft Visual Studio Debug Console
Matrix 3x3 criada:
0 0 0
0 0 0
0 0 0

Matriz Diagonal com 2:
2 0 0
0 2 0
0 0 2
Matriz diagonal?: S

Matriz nao Diagonal:
2 0 0
0 2 6
0 0 2
Matriz diagonal?: N
Memoria liberada
```

¹ Nesse exemplo, você tem um vetor de ponteiros: `int **v`, onde `v[0]`, `v[1]` e `v[2]` são ponteiros para `int`. Observe que você tem que primeiro alocar 3 espaços de memória do tamanho de `int *`, e depois, cada `v[i]` deve guardar o endereço de uma área de memória alocada com espaço suficiente para conter 3 números inteiros.

² Uma matriz é diagonal se ela é quadrada e se todos os elementos fora da diagonal são zeros.