

...  
Arquivo: matriz\_palavras\_cruzadas.py

---

Um jogo de palavras cruzadas pode ser representado por uma matriz  $a$  com  $m$  linhas e  $n$  colunas, onde cada posição da matriz corresponde a um quadrado do jogo, sendo que  $0$  representa um quadrado branco e  $-1$  representa um quadrado preto.

Indicar na matriz as posições que são inícios de palavras na horizontal ou na vertical (substituindo os zeros), considerando que uma palavra deve ter pelo menos duas letras.

Para isso, numere consecutivamente tais posições. (na ordem das linhas)

Exemplo: Para  $m=5$ ,  $n=8$  e a matriz dada a seguir,

0	-1	0	-1	-1	0	-1	0
0	0	0	0	-1	0	0	0
0	0	-1	-1	0	0	-1	0
-1	0	0	0	0	-1	0	0
0	0	-1	0	0	0	-1	-1

a saída deverá ser:

1	-1	2	-1	-1	3	-1	4
5	6	0	0	-1	7	0	0
8	0	-1	-1	9	0	-1	0
-1	10	0	11	0	-1	12	0
13	0	-1	14	0	0	-1	-1

...

#-----

(A)

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	0	-1	0	-1	-1	0	-1	0	-1	-1
-1	0	0	0	0	-1	0	0	0	-1	-1
-1	0	0	-1	-1	0	0	-1	0	-1	-1
-1	-1	0	0	0	0	-1	0	0	-1	-1
-1	0	0	-1	0	0	0	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

a saída deverá ser:

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	2	-1	-1	3	-1	4	-1	-1
-1	5	6	0	0	-1	7	0	0	-1	-1
-1	8	0	-1	-1	9	0	-1	0	-1	-1
-1	-1	10	0	11	0	-1	12	0	-1	-1
-1	13	0	-1	14	0	0	-1	-1	-1	-1

**A**

```
def main():
    nlin = int(input("Digite o número de linhas da matriz: "))
    ncol = int(input("Digite o número de colunas da matriz: "))

    # criar a 'estrutura' para a matriz a com moldura
    a = cria_matriz(nlin+2, ncol+2, -1) ←
    # as linhas 0 e nlin+1, e as colunas 0 e ncol+1 formam uma moldura

    for i in range(1, nlin+1): ←
        for j in range(1, ncol+1): ←
            a[i][j] = int(input("Digite o elemento [%d][%d]: " %(i, j)))

    print("\nMatriz com uma moldura:")
    imprime_matriz(a)
```

**B**

```
numera_matriz(a) ← chamada da função que faz
                    a numeração
```

```
print("\nMatriz numerada com uma moldura:")
imprime_matriz(a)
```

```
print("\nMatriz com a numeração de inícios de palavras na horizontal"
      " ou na vertical:")
print("\nMatriz com %d linhas e %d colunas\n" %(nlin, ncol))

for i in range(1, nlin+1): ←
    for j in range(1, ncol+1): ←
        print("%6d" %(a[i][j]), end=' ')
    print()
print()
```

```
def imprime_matriz(matriz):
    ''' (matriz) -> NoneType

    Recebe uma matriz de inteiros e imprime-a no formato
    bidimensional de matriz.
    '''

    nlin = len(matriz)
    ncol = len(matriz[0])

    print("\nMatriz com %d linhas e %d colunas\n" %(nlin, ncol))

    for i in range(0, nlin):
        for j in range(0, ncol):
            print("%6d" %(matriz[i][j]), end=' ')
        print()
    print()
```

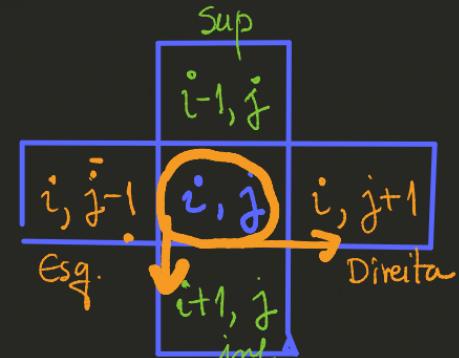


```
def numera_matriz(matriz):  
    ''' (matriz) -> NoneType
```

Recebe uma matriz e numera consecutivamente as posições que podem ser início de palavras na horizontal ou na vertical. Essas palavras devem ter comprimento pelo menos dois.  
Obs. A função supõe que a matriz tem apenas os inteiros 0 ou -1 e que ela tem uma moldura com -1.

```
...  
  
nlin = len(matriz) - 2  
ncol = len(matriz[0]) - 2
```

```
num = 1  
for i in range(1, nlin+1):  
    for j in range(1, ncol+1):  
        if matriz[i][j] == 0:  
            if matriz[i][j-1] == -1 and matriz[i][j+1] == 0:  
                # é início de uma palavra na horizontal  
                matriz[i][j] = num  
                num += 1  
            elif matriz[i-1][j] == -1 and matriz[i+1][j] == 0:  
                # é início de uma palavra na vertical  
                matriz[i][j] = num  
                num += 1
```



..

(Vejá \*)

```
def cria_matriz(nlin, ncol, valor):
    ''' (int, int, tipo do valor) -> matriz (ou seja, tipo
```

Cria uma matriz com nlin linhas e ncol colunas,  
sendo que cada elemento é igual a valor.  
Retorna a matriz criada.

```
'''
```

```
matriz = []
for i in range(0, nlin):
    linha = []
    for j in range(0, ncol):
        linha.append(valor)
    matriz.append(linha)

return matriz
```

Dica p/ EP4

```
#-----  
main()
```

\*

lista\_posicoes = [ ]

lista\_posicoes.append( (i,j) )

Dica API EP4

[ lista\_posicoes=[(i,j)] ]

fila