

ESCOLA
SUPERIOR
DE MEDIA
ARTES
E DESIGN
POLITÉCNICO
DO PORTO



```
Algoritmia e Estruturas de Dados

UNIDADE CURRICULAR

Ficha 07 – Listas unidimensionais e bidimensionais
```

 Dada uma matriz 3x3 de inteiros, cujos valores são indicados pelo utilizador, elabore uma função invert que receba a matriz lida e imprima, assim como à sua transposta.

```
C:\WINDOWS\py.exe
Linha 1, coluna 1
Linha 1, coluna 2
Linha 1, coluna 3
Linha 2, coluna 1 : 4
Linha 2, coluna 2 : 5
Linha 2, coluna 3
Linha 3, coluna 1 : 7
Linha 3, coluna 2 : 8
Linha 3, coluna 3 : 9
Matriz Original:
1 2 3
4 5 6
7 8 9
Matriz Transposta:
1 4 7
2 5 8
```

2. Elabore um programa com o seguinte menu inicial:

```
MENU
1 - Inicializar matriz
2 - Matriz Transposta
3 - maior valor
0 - Sair
Escolha uma das opções: _
```

Inicializar matriz: deve pedir ao utilizador a dimensão da matriz (nº de linhas /co-lunas da matriz) e invocar uma função que preencha a matriz com valores aleatórios entre 10 e 100. A função deve imprimir a matriz gerada.



```
MENU

1 - Inicializar matriz

2 - Matriz Transposta

3 - maior valor

0 - Sair
Escolha uma das opções: 1

Dimensão da matriz:3

Matriz Gerada:
[18, 64, 74]
[80, 15, 76]
[99, 77, 74]
```

 Matriz transposta: deve invocar uma função que <u>receba a matriz gerada</u>, e imprima a sua transposta

```
MENU

1 - Inicializar matriz

2 - Matriz Transposta

3 - maior valor

0 - Sair
Escolha uma das opções: 2

Matriz Original:

18 64 74

80 15 76

99 77 74

Matriz Transposta:

18 80 99

64 15 77

74 76 74
```

 Maior valor: deve invocar uma função <u>que receba a matriz</u> e imprima o maior valor da matriz

```
MENU

1 - Inicializar matriz

2 - Matriz Transposta

3 - maior valor

0 - Sair

Escolha uma das opções: 3

0 maior valor da matriz é 99
```

• Sair : deve terminar o seu programa



3. Elabore um programa que permita gerir a ocupação de um pequeno parque de estacionamento, com o *layout* abaixo apresentado (3 filas de estacionamento, cada uma delas com 5 lugares).

1	2	3	4	5	Fila 1
1	2	3	4	5	Fila 2
1	2	3	4	5	Fila 3

Quando o programa se inicia, todos os lugares do parque devem estar livres. O seu programa deve conter um menu com as seguintes opções:

MENU

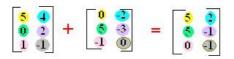
- 1 Entrada de veículo
- 2 Saída de carro
- 3 Estado do Parque
- 0 Sair
- Entrada de veículo: deve ocupar o primeiro lugar que estiver livre, começando pela Fila 1 e terminando na Fila 3. Deve indicar na consola a posição do lugar a ocupar.

Se todos os lugares estiverem ocupados deverá surgir a mensagem de "Parque completo".

- Saída de carro: O utilizador deve indicar a posição, na fila de estacionamento, do carro que pretende sair (fila e lugar). Esse lugar deve passar ao estado de livre.
- **Estado do Parque:** esta opção deve indicar, na consola, o número de lugares ocupados e o número de lugares livres, no parque.
- 4. Implemente um programa que permita somar e subtrair duas matrizes (devem ter a mesma ordem, isto é, as duas matrizes devem ter igual número de linhas e colunas).
- Somar duas matrizes:

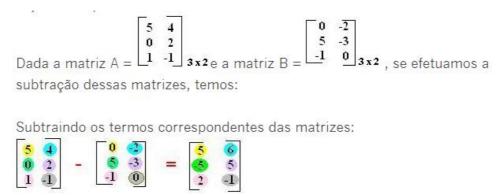
Dado a matriz A =
$$\begin{bmatrix} 5 & 4 \\ 0 & 2 \\ 1 & -1 \end{bmatrix}$$
 3×2 e matriz B = $\begin{bmatrix} 0 & -\overline{2} \\ 5 & -3 \\ -1 & 0 \end{bmatrix}$ 3×2 , se efetuarmos a soma dessas matrizes teremos:

Somaremos os termos correspondentes em cada matriz:





• Subtrair duas matrizes:



Exemplo de menu pretendido e da *interface* associada à soma de matrizes. A *interface* associada à subtração de matrizes deverá ter um *layout* semelhante à apresentada para a soma.

