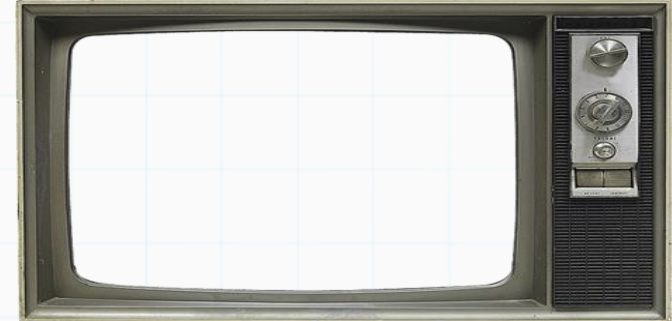


Programação De Computadores

Professor : Yuri Frota

www.ic.uff.br/~yuri/prog.html

yuri@ic.uff.br



Usar apenas comandos de listas vistos na aula:

append (inserir elemento no final da lista)

len (tamanho da lista)

+ (concatenação de listas)

***** (multiplicação de listas)



Velha

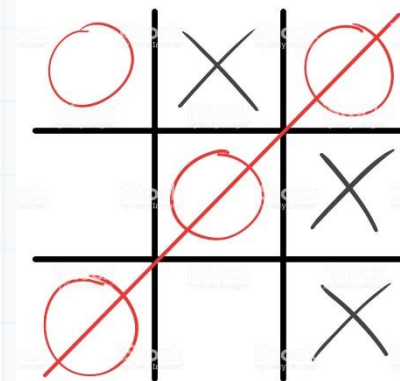
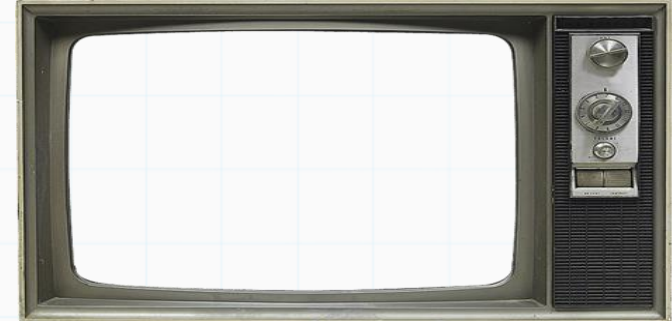
Exercício 1): Implemente o jogo da velha para ser jogado com 2 jogadores. Para esse exercício você deve completar o arquivo do programa velha.py que implementa o tabuleiro como uma matriz 3x3.

<http://www.ic.uff.br/~yuri/python/velha.py>

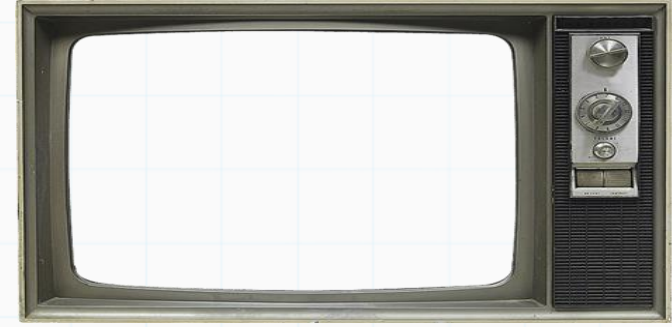
programa deve ter as seguintes funcionalidades:

- a) marcar a jogada e proibir jogadas irregulares (espaços já preenchidos ou fora do tabuleiro).
- b) alternar as jogadas entre os jogadores 1 e 2
- c) testar as condições de vitória a cada jogada
- d) testar empate

A seguir temos um exemplo de como seria a execução do programa. Será que você consegue implementar o joguinho usando tudo que aprendeu na disciplina ?



Velha (exemplo de execução)



```
-----  
[' ', ' ', ' ', ' ']  
[' ', ' ', ' ', ' ']  
[' ', ' ', ' ', ' ']  
-----
```

Jogador 1:
linha :1
coluna:1

Jogador 1:
linha :0
coluna:1

```
-----  
['O', 'X', ' ', ' ']  
[' ', 'X', ' ', ' ']  
[' ', ' ', ' ', ' ']  
-----
```

Jogador 2:
linha :0
coluna:2

```
-----  
['O', 'X', 'O']  
[' ', 'X', ' ']  
['X', 'O', ' ']  
-----
```

```
-----  
[' ', ' ', ' ', ' ']  
[' ', 'X', ' ', ' ']  
[' ', ' ', ' ', ' ']  
-----
```

Jogador 2:
linha :0
coluna:0

Jogador 2:
linha :2
coluna:1

```
-----  
['O', 'X', ' ', ' ']  
[' ', 'X', ' ', ' ']  
[' ', 'O', ' ', ' ']  
-----
```

Jogador 1:
linha :2
coluna:2

```
-----  
['O', 'X', 'O']  
[' ', 'X', ' ']  
['X', 'O', 'X']  
-----
```

Jogador 1:
linha :1
coluna:0

```
-----  
['O', 'X', 'O']  
['X', 'X', 'O']  
['X', 'O', 'X']  
-----
```

deu velha

```
-----  
['O', ' ', ' ', ' ']  
[' ', 'X', ' ', ' ']  
[' ', ' ', ' ', ' ']  
-----
```

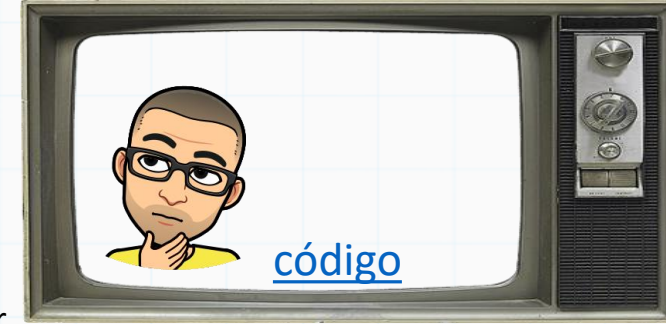
Jogador 1:
linha :2
coluna:0

```
-----  
['O', 'X', ' ', ' ']  
[' ', 'X', ' ', ' ']  
['X', 'O', ' ', ' ']  
-----
```

Jogador 2:
linha :1
coluna:2

```
-----  
['O', 'X', 'O']  
[' ', 'X', 'O']  
['X', 'O', 'X']  
-----
```

Matrizes Cont.- LAB



Exercício 1.2): Receba uma matriz A $n \times m$ de inteiros e ordene seus elementos de forma que os menores fiquem na primeira linha, depois os próximos menores na segunda linha e assim por diante (e cada linha também esteja ordenada).

Ex:

4	3	5
3	0	6
5	7	4

 →

0	3	3
4	4	5
5	6	7

Dica: Crie um vetor (lista) de tamanho $n \times m$, com os elementos da matriz, ordene, e jogue de volta na matriz.

4	3	5	3	0	6	5	7	4
---	---	---	---	---	---	---	---	---

 →

0	3	3	4	4	5	5	6	7
---	---	---	---	---	---	---	---	---

matriz -> vetor

I percorre as linhas da matriz

J percorre as colunas da matriz

insere elementos no vetor de ordenação

Ordena vetor

I percorre o vetor

J percorre o vetor

Varredura: troca as posições J e J+1 que não estiverem de acordo com a ordenação

vetor -> matriz

I percorre as linhas da matriz

J percorre as colunas da matriz

atualiza posição da matriz com a próxima posição do vetor

Matrizes Cont.

Exercício 1.5): Receba uma matriz de inteiros A de dimensão $n \times m$, em seguida ordene cada coluna da matriz da seguinte forma.

Se a coluna tiver índice par -> ordem crescente

Se a coluna tiver índice ímpar -> ordem decrescente

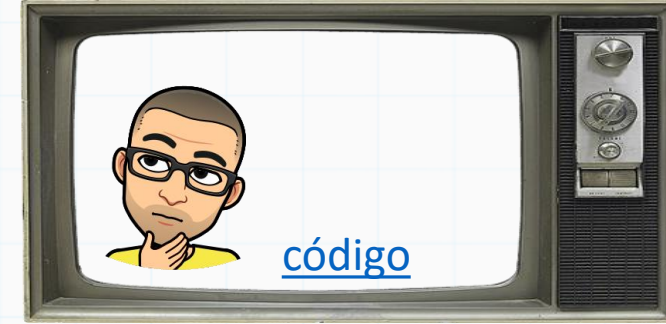
Imprima a matriz após a ordenação.

Ex: $n=3$ $m=4$

0)	1)	2)	3)
3	1	7	0
0	9	6	7
5	4	11	0



0	9	6	7
3	4	7	0
5	1	11	0



K percorre colunas da matriz

Se K for par então

I percorre linhas da matriz

J percorre linhas da matriz

Varredura: troca as posições J e J+1 da coluna K que não estiverem de acordo com a ordenação crescente

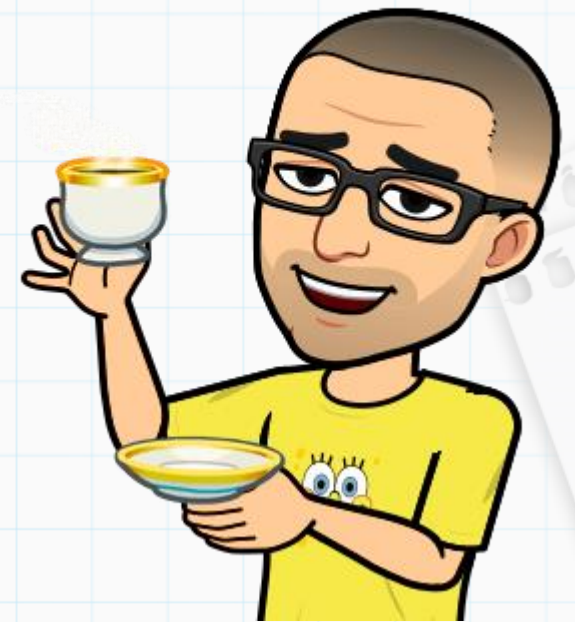
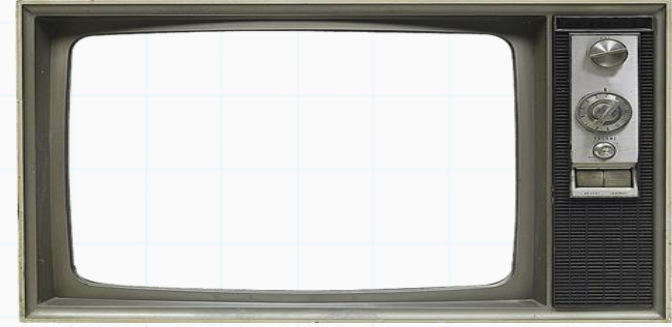
Se K for ímpar então

I percorre linhas da matriz

J percorre linhas da matriz

Varredura: troca as posições J e J+1 da coluna K que não estiverem de acordo com a ordenação decrescente

Até a próxima



Slides baseados no curso de Vanessa Braganholo