

---

## QUESTÃO DISCURSIVA – DRAGON STONE

---

*Dragon Stone é uma ilha na região de Westeros onde diversos dragões vivem. A ilha funciona como uma espécie de parque, onde os dragões permanecem em cavernas escuras. A qualquer tempo, seus montadores podem requisitá-los, seja para viagens rápidas, seja para a guerra.*

Considere a classe Dragão disponibilizada na questão anterior e faça o que é pedido.

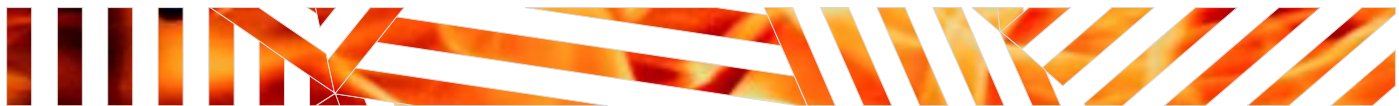
- a) Crie um construtor para a classe Dragão. Esse construtor deverá receber três parâmetros (conforme é mostrado na assinatura do método abaixo) e inicializar os respectivos atributos.

```
public Dragao(string nome, string montador, int anoNascimento)
```

- b) Com base na função Main (arquivo Program.cs) indicado, faça o que é pedido:

- 1) Crie um novo objeto de Dragão. Atribua a seus atributos valores lidos do console (o usuário que irá digitar). Insira esse novo objeto na lista de dragões.
- 2) Percorra a lista de dragões e imprima na tela a descrição de cada objeto.
- 3) Imprima na tela o nome dos dragões precedido do índice dele na lista. Em seguida, pergunte ao usuário qual dragão ele deseja remover. O usuário irá inserir um número inteiro e, se esse número for válido, o dragão nesta posição será removido da lista.





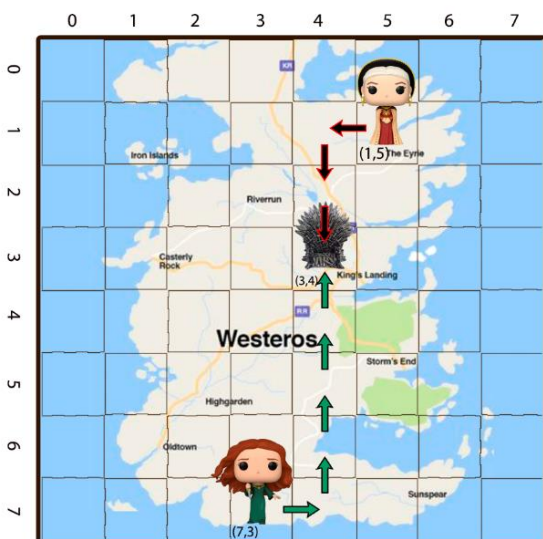
## QUESTÃO DISCURSIVA – HEIR TO THE IRON THRONE



Os apoiadores de Aegon Targaryen, filho de Alicent Hightower, entram em conflito com os de sua meia-irmã, Rhaenyra Targaryen, pelo trono de Viserys I, seu falecido pai, iniciando uma guerra civil conhecida como a Dança dos Dragões. Dois grupos distintos conhecidos como Greens e Blacks liderados por Alicent e Rhaenyra, respectivamente, se enfrentam para assumir a posição mais cobiçada do reino, o trono de ferro.

Considere a região de Westeros como um tabuleiro de xadrez (dimensões 8 x 8) conforme a figura abaixo. Cada linha (x) e cada coluna (y) pode ser indexada com um inteiro entre 0 e 7. A posição do Trono de Ferro, dos Greens e dos Blacks são dadas por (x,y), simbolizando Peças no tabuleiro. Cada Peça pode se mover nas direções  $\uparrow, \leftarrow, \downarrow, \rightarrow$  ao passo que cada movimento é considerado uma unidade de distância percorrida.

Observe o exemplo:



Temos que o Trono de Ferro está na posição (3,4), os Greens na posição (7,3) e os Blacks na posição (1,5).

- Para que os Greens cheguem até o trono, é necessário realizar no mínimo 5 movimentos (direita ( $\rightarrow$ ), cima ( $\uparrow$ ), cima ( $\uparrow$ ), cima ( $\uparrow$ ) e cima ( $\uparrow$ )).
- Para que os Blacks cheguem ao trono é necessário realizar no mínimo 3 movimentos (esquerda ( $\leftarrow$ ), baixo ( $\downarrow$ ) e baixo ( $\downarrow$ )).

Considerando o que foi dito, uma classe denominada **Peça** foi implementada visando representar elementos no tabuleiro. Observe o código da classe Peça e faça o que é pedido.

```
class Peça
{
    public int X;
    public int Y;
    public string Nome;
}
```

Adicione no seu projeto de Console a classe Peça conforme mostrado ao lado. Adicione também o método de assinatura abaixo na classe:

```
public static int Distância(Peça p1, Peça p2).
```

Essa função deve receber dois objetos de Peça (p1 e p2 já instanciados) e retornar a distância entre essas duas peças no tabuleiro.

Após implementar o que foi pedido, execute o seguinte código na Main para validar o que foi feito. Sua saída deve ser 5 e 3.

```
Peça greens = new Peça();
greens.X = 7; greens.Y = 3; greens.Nome = "Alicent Hightower";
Peça blacks = new Peça();
blacks.X = 1; blacks.Y = 5; blacks.Nome = "Rhaenyra Targaryen";
```

```
Peça throne = new Peça();
throne.X = 3; throne.Y = 4; throne.Nome = "Trono de Ferro";
Console.WriteLine(Peça.Distância(greens, throne));
Console.WriteLine(Peça.Distância(blacks, throne));
```





--	--

