**2a Prova Introdução à Programação**

**Prof Alexandre Mota – 14/11/2017**

**Questão**. Em um jogo de xadrez, temos as seguintes peças (com respectivos movimentos permitidos):

* **Peão** (8 unidades de cada cor): Apenas uma casa para frente, se não houver peça. Pula peça (de outra cor) em movimento diagonal (2 casas), e a peça pulada é removida;
* **Torre** (2 unidades de cada cor): Movimentos verticais ou horizontais, mas não pode haver peça da mesma cor no caminho. Havendo (de outra cor), sua torre assume a posição desta peça e esta é removida do jogo;
* **Bispo** (2 unidades de cada cor): Movimentos diagonais e não pode haver peças de mesma cor no caminho. Bispo assume posição de peça de outra cor e esta sai do jogo;
* **Cavalo** (2 unidades de cada cor): Movimento em L (2 casas pra frente/trás e 1 uma pra um dos lados ou 1 casa pra frente/trás e 2 pra um dos lados). Pode passar por sobre qualquer peça. Se o local, onde cavalo pára, tiver peça de outra cor, esta sai do jogo. Se mesma cor, jogada inválida;
* **Rei** (1 unidade de cada cor): Qualquer destino é possível desde que só movimente uma célula por vez. Não pode haver peça de mesma cor no caminho;
* **Rainha** (1 unidade de cada cor): Qualquer destino é possível e qualquer quantidade de células. Não pode haver peça de mesma cor no caminho.

Que são dispostas conforme o tabuleiro a (esquerdo):

|  |  |  |
| --- | --- | --- |
| Macintosh HD:Users:alexandremota:Desktop:Captura de Tela 2017-11-13 às 18.38.27.png | E algumas jogadas depois... |  |

Os movimentos das peças, intercalados entre brancas e pretas, uma cor (jogada) de cada vez, são capturados por três principais informações:

1. Um caractere (**T**–Torre, **C**–Cavalo, **B**– Bispo, **D**– Dama/Rainha, **R**– Rei). Apesar de não tradicional, considere **P** para peão;
2. Posição de origem da peça;
3. Posição de destino da peça.

Um exemplo de jogada acima é **Bf1-c4**, a qual desloca o bispo da casa **f1** para **c4**.

Pede-se que você crie um programa em C que registre as jogadas em 2 vetores dinâmicos, onde cada jogada deve ser validada antes, alertando o usuário caso não seja válida. **Crie funções auxiliares para tais verificações**. Mostre um novo tabuleiro (matriz) na tela do computador a cada jogada. Ao final do jogo (rei foi capturado), seu programa deve armazenar a partida em um arquivo binário de nome **partida.bin**.

Boa sorte!!!