

Trabalho Prático 3 - Damas (Lei da maioria)

Em 1756, Samuel Johnson redigiu o que se tornaria o primeiro livro a mencionar o nome do jogo: damas (em inglês “draughts”). Atualmente, existem três variações de tabuleiro e número inicial de peças: 8x8 com 12 peças cada (origem inglesa e russa, também comum no Brasil e EUA), 10x10 com 20 peças cada (versão internacional e mais popular) e 12x12 com 30 peças cada (versão canadense). Na Figura 1, vemos um exemplo de tabuleiro de damas brasileiro com 64 casas, cujos dois jogadores iniciam com 12 peças cada. Essas peças são posicionadas sobre as casas escuras de forma que 4 delas tenham na diagonal esquerda uma casa livre para se movimentar.

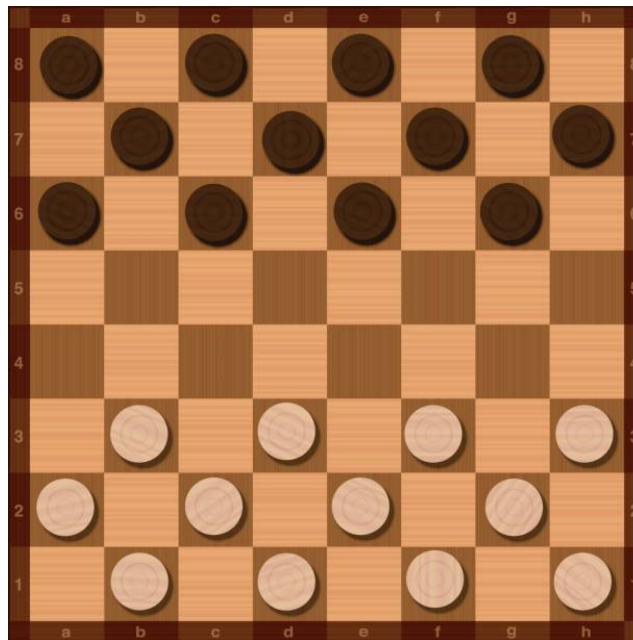


Figura 1: Tabuleiro 8x8 com disposição inicial de peças

De um lado, o jogador deve escolher qual peça irá se mover e para onde (sempre na diagonal). Quando uma peça de um jogador se encontra em uma casa vizinha a uma peça do outro jogador, ele deve obrigatoriamente “capturar” a peça (saltando-a e posicionando-se na próxima casa no sentido transversal ao seu movimento). Se, logo após capturar uma peça, o jogador puder capturar outra peça, então ele deverá fazê-lo na mesma jogada (capturando em sequência quantas peças forem possíveis). Essa regra é conhecida pelo nome “lei da maioria”.

O trabalho prático consiste na implementação de um algoritmo que dará assistência ao comitê brasileiro de damas. Esse algoritmo irá poupar os jogadores de identificarem se há um movimento obrigatório e qual ele seria (seguindo a “lei da maioria”). Em outras palavras, seu

algoritmo deve identificar todas **oportunidades de captura** e eleger aquela que resultará **no maior número de peças capturadas**. Para validação do resultado, o comitê requisitou que, se o programa identificasse múltiplas peças com ao menos uma **oportunidade de captura**, então ele deveria listá-las juntamente ao número de capturas em seu melhor movimento.

Dado um tabuleiro de damas com peças dispostas de uma forma informada e a cor das peças que devem se mover na atual rodada, sua implementação deverá ser capaz de:

1. Informar a posição inicial da peça que deverá se mover segundo a “lei da maioria”;
2. Informar o número de peças que serão capturadas nesse movimento;
3. Seguindo a ordem em que as peças estão dispostas no tabuleiro, percorrendo-o como ilustrado na Figura 2, avaliar quais peças tem ao menos uma **oportunidade de captura**, incluindo a peça que irá se mover, e:
 - a. Informar suas posições iniciais no tabuleiro;
 - b. Informar o número máximo de capturas que elas serão capazes de fazer em uma rodada;

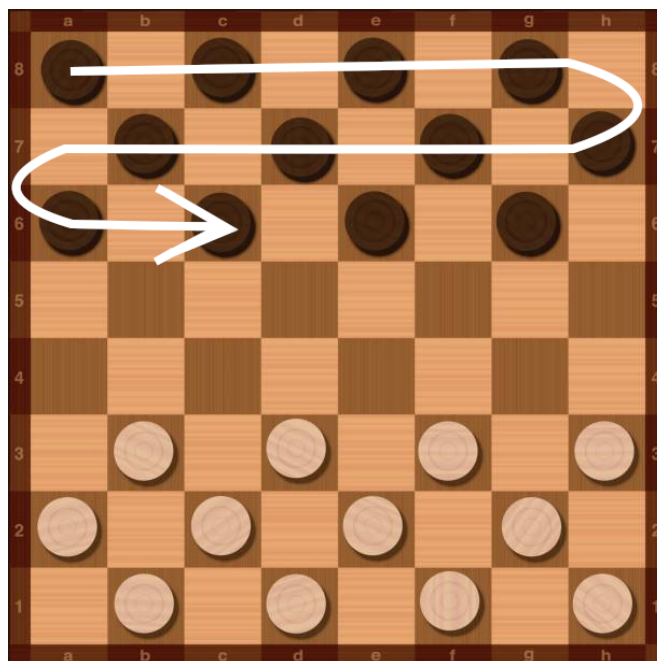


Figura 2: Tabuleiro com seta sobreposta indicando ordem de avaliação de peças

O programa inicialmente deve ler as entradas contendo informações gerais sobre o tabuleiro do jogo, como dimensões (8x8, 10x10 ou 12x12) e posição das peças, e a cor das peças que deverão se mover. Em seguida, o programa deve descrever o movimento eleito pela “lei da maioria” e todas as **melhores oportunidades de captura** de cada peça.

ATENÇÃO: Seu programa deverá utilizar recursividade para avaliar todos movimentos válidos enquanto for possível efetuar capturas. Siga as regras da versão internacional de damas: na qual qualquer peça pode efetuar captura **para trás ou para frente**. E considere que dentre as peças dispostas no tabuleiro **não há** damas.

ENTRADA: A entrada possui várias linhas, onde cada linha tem detalhes sobre o jogo.

- A primeira linha possui um valor inteiro N , que indica o tamanho do tabuleiro em linhas e colunas ($X \in \{8, 10, 12\}$). ATENÇÃO: o tabuleiro possui casas brancas e pretas e somente as casas pretas são usadas (veja exemplos);
- As próximas N linhas conterão N caracteres cada, sendo que o i -ésimo caractere da j -ésima linha corresponde a casa na posição a_{ij} . Os possíveis valores de a_{ij} são: “o” (casa preta vazia), “p” (casa preta com peça preta), “b” (casa preta com peça branca) e “ ” (casa branca).

SAÍDA: Ao fim da execução, o programa deverá imprimir as seguintes informações:

1. Qual peça deverá se mover segundo a “lei da maioria” e quantas peças serão capturadas durante o movimento. Representado em uma linha contendo três números inteiros, x y n , sendo
 - n , o número de capturas,
 - x e y , as coordenadas da posição da peça no tabuleiro:
 - x , indicando o eixo horizontal crescendo no sentido original (da esquerda para a direita)
 - y , indicando o eixo vertical crescendo no sentido inverso (de cima para baixo).
2. Quais as demais peças com oportunidade de captura e, no máximo, quantas capturas eram capazes de fazer em apenas um movimento. Representado em m linhas, sendo m o número de peças com oportunidade de captura (incluindo a peça que deverá se mover), contendo três números inteiros, x y n , sendo
 - n , o número de capturas,
 - x e y , as coordenadas da posição inicial da peça no tabuleiro:
 - x , indicando o eixo horizontal crescendo no sentido original (da esquerda para a direita)
 - y , indicando o eixo vertical crescendo no sentido inverso (de cima para baixo).

ENTREGA: Cópias de trabalho são inaceitáveis, podem ser detectadas automaticamente e serão repassadas ao conselho disciplinar. Seu código deve ser desenvolvido na linguagem C e entregue através do Moodle.

EXEMPLOS:

Entrada	Saída
12 o o o o o o o p p p o p o o o o b o o p p o p p o o o o b o o p p p o p o o o o p o o p p p b o o o o b p p o p p b o b o b o o o b b b b b b b	10 2 10 2 8 7 4 8 9 8 6 7 10 2 10
8 p p p p p p p p p p o p o o p o o o b b b b o o b b b b b b b b	4 6 1 4 6 1
10 o o o o o o p p p o o o b o o o p o p o o b o b o o o o p o o o o b o o o o o o o o o o o o o o o o	6 6 5 2 4 2 4 6 2 6 6 5
8 b b o b p p p o b o b o p p p o o b o o p p p o b o b o o o o o	0 0 9 0 0 9 2 4 1 6 4 1

DICA: Você não precisará usar ordenação, ao invés disso, basta usar [malloc](#) e [realloc](#) para adicionar os inteiros **x y n** a um vetor (na ordem que lê-los).