Terceiro Exercício-Programa

Prof. Luciano Antonio Digiampietri

Prazo máximo para a entrega: 02/07/2021

1 Jogo da Velha

O Jogo da Velha, Jogo do Galo ou Jogo das Três Linhas é um jogo bastante popular com regras simples. Ele é jogado por dois jogadores, sendo composto por um tabuleiro com nove casas, arranjadas em três linhas e três colunas. O tabuleiro é iniciado vazio, e em cada jogada um jogador coloca uma "peça" em uma das casas vazias. Vamos considerar que o primeiro jogador possui a peça X (representado neste EP pela letra 'x' maiúscula) e o segundo jogador possui a peça O (representado neste EP pela letra 'o' maiúscula).

Consideraremos que o jogo sempre será iniciado pelo jogador com a peça X. O jogo acaba assim que um dos jogadores conseguir colocar três de suas peças (três X ou três O) em uma sequência: na mesma linha, na mesma coluna ou em uma das diagonais. Neste caso, o jogador que conseguiu produzir essa sequência será o ganhador. A outra alternativa para o fim do jogo é as nove casas do tabuleiro serem preenchidas com X e O sem nenhuma das sequências vitoriosas acontecerem. Neste caso, o jogo acabará em empate[1][2].

Neste EP você deverá implementar uma função que verifique o *status* de um tabuleiro válido durante um Jogo da Velha. Os possíveis estados do tabuleiro são:

- 0 – Jogo não iniciado: o tabuleiro está "vazio", isto é sem peças X e O;
- 1 – Jogo encerrado 1: o primeiro jogador (que usa as peças X) é o ganhador;
- 2 – Jogo encerrado 2: o segundo jogador (que usa as peças O) é o ganhador;
- 3 – Jogo encerrado 3: empate – todas as casas do tabuleiro estão preenchidas com X e O, mas nenhum dos jogadores ganhou;
- 4 – Jogo já iniciado e em andamento: nenhuma das alternativas anteriores.

Você deverá implementar uma função com a seguinte assinatura:

```
int verificaStatus(char tabuleiro[][3])
```

[1] `https://pt.wikipedia.org/wiki/Jogo_da_velha`
[2] `https://en.wikipedia.org/wiki/Tic-tac-toe`

Esta função receberá um tabuleiro no formato de uma matriz 3×3 de caracteres. Você pode assumir que todos os tabuleiros sempre corresponderão a jogos válidos (isto é, sempre estarão preenchidos com X, O ou espaço em branco (caractere `' '`, e não o caractere vazio `''`, para representar casas vazias/livres); os jogadores alternarão entre suas jogadas e não é possível em um único jogo haver dois ganhadores.

1.1 Entrada

A entrada é composta pelo parâmetro da função, que recebe uma matriz de caracteres com dimensões 3×3, correspondendo a um tabuleiro de jogo da velha.

1.2 Saída

A função deverá retornar um número inteiro (entre 0 (zero) e 4, correspondendo ao *status* do jogo do tabuleiro atual, conforme a lista de estados possíveis já apresentada).

Para que você teste o funcionamento da função `verificaStatus`, a função `main()` traz alguns testes para diferentes tabuleiros de entrada, mas você pode/deve realizar outros testes. No código fornecido juntamente com o enunciado já são apresentados alguns exemplos (se desejar imprimir coisas na tela, faça isso **apenas na função main** durante seus testes; **não use `printf` diretamente em `verificaStatus`**).

1.3 Material a Ser Entregue

Um arquivo, denominado *`NUSP`*`.c` (sendo NUSP o seu número USP, por exemplo: 123456789.c), contendo seu código com a função `verificaStatus` e qualquer outra função adicional que ache necessário. Para sua conveniência, `completeERenomeie.c` será fornecido, cabendo a você então completá-lo e renomeá-lo para a submissão.

Atenção!

1. Não modifique a assinatura de `verificaStatus`!
2. Para avaliação, apenas a função `verificaStatus` será invocada diretamente. Em especial, qualquer código dentro da função `main()` será ignorado. Então certifique-se de que o problema seja resolvido chamando-se diretamente somente essa função.

2 Entrega

A entrega será feita única e exclusivamente via sistema e-Disciplinas, até a data final marcada. Deverá ser postado no sistema um arquivo c, tendo como nome seu número USP:

```
seuNumeroUSP.c   (por exemplo, 12345678.c)
```

Não esqueça de preencher o cabeçalho constante do arquivo .c, com seu nome, número USP e turma etc.

A responsabilidade de postagem é exclusivamente sua. Por isso, submeta e certifique-se de que o arquivo submetido é o correto (fazendo seu download, por exemplo). Problemas referentes ao uso do sistema devem ser resolvidos com antecedência.

3 Avaliação

A nota atribuída ao EP terá como foco principal a funcionalidade solicitada, porém não esqueça de se atentar aos seguintes aspectos:

1. Documentação: se há comentários explicando o que se faz nos passos mais importantes e para que serve o programa (Tanto a função quanto o programa em que está inserido);
2. Apresentação visual: se o código está legível, identado etc;
3. Corretude: se o programa funciona.

Além disso, algumas observações pertinentes ao trabalho, que influem em sua nota, são:

- Este exercício-programa deve ser elaborado individualmente;
- Não será tolerado plágio, em hipótese alguma;
- Exercícios com erro de sintaxe (ou seja, erros de compilação), receberão nota ZERO.