UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Bacharelado em Sistemas de Informação
Técnicas de Programação I - 02/2020
Prof. Jobson Massollar

Trabalho 2

LEIA ATENTAMENTE AS REGRAS E OS ENUNCIADOS

R E G R A S

- O trabalho deverá ser realizado individualmente.
- O trabalho deverá ser enviado para o Google classroom até o dia 23/04/2021 (sexta-feira).
- A data de entrega não será adiada.
- Os 3 programas solicitados (arquivos .CPP) deverão ser compactados em um único arquivo (ZIP ou RAR) com o nome e sobrenome do aluno.
- Os programas (arquivos .CPP) deverão ter os nomes conforme definido nos enunciados.
- Não serão aceitos trabalhos enviados por email.
- Trabalhos com estruturas e/ou organizações semelhantes (plágio) serão penalizados com a nota zero.
- O programa que não obedecer às restrições estabelecidas receberá zero.

E N U N C I A D O S

1) Programa: jogodavelha.cpp (6,0 pontos)

Criar um jogo da velha onde cada jogador (jogador 1 e jogador 2) vai, alternadamente, informar a coordenada (linha e coluna) que ele deseja marcar. O jogador 1 vai usar um X para marcar suas posições e o jogador 2 vai usar um O. Posições vagas devem ser marcadas com _ (underscore). O programa termina quando um dos jogadores vence (o programa deverá imprimir quem venceu) ou quando o jogo termina empatado (velha). Antes de cada jogador realizar sua jogada o programa deve imprimir as nove posições do cenário atual do jogo em formato de matriz 3 x 3.

Restrições:

a) O jogador 1 sempre inicia o jogo.
b) As coordenadas informadas pelos jogadores deverão estar na ordem **linha coluna**, onde linha e coluna variam de 1 a 3. Caso linha ou coluna não estejam no intervalo de 1 a 3, o programa deverá apresentar uma mensagem de erro e solicitar a coordenada novamente.

c) Uma posição já ocupada não pode ser ocupada novamente. Caso isso ocorra, o programa deverá apresentar uma mensagem de erro e solicitar a coordenada novamente.
d) Quando restar apenas uma posição a ser preenchida, o programa deverá simular a jogada do próximo jogador e informar se ele vence ou deu empate (velha).

Exemplos de execução do jogo da velha:

Exemplo 1

```
_  _  _
_  _  _
_  _  _
Jogador 1: 1 1
X  _  _
_  _  _
_  _  _
Jogador 2: 2 2
X  _  _
_  O  _
_  _  _
Jogador 1: 3 1
X  _  _
_  O  _
X  _  _
Jogador 2: 2 1
X  _  _
O  O  _
X  _  _
Jogador 1: 2 3
X  _  _
O  O  X
X  _  _
Jogador 2: 3 2
X  _  _
O  O  X
X  O  _
Jogador 1: 1 2
X  X  _
O  O  X
X  O  _
Jogador 2: 1 3
X  X  O
O  O  X
X  O  _
Velha!
```

Exemplo 2

```
_  _  _
_  _  _
_  _  _
Jogador 1: 1 3
_  _  X
_  _  _
_  _  _
Jogador 2: 2 4
Coordenadas invalidas!
Jogador 2: 2 2
_  _  X
_  O  _
_  _  _
Jogador 1: 3 1
_  _  X
_  O  _
X  _  _
Jogador 2: 1 1
O  _  X
_  O  _
X  _  _
Jogador 1: 2 2
Posição já ocupada!
Jogador 1: 3 3
O  _  X
_  O  _
X  _  X
Jogador 2: 3 2
O  _  X
_  O  _
X  O  X
Jogador 1: 2 3
O  _  X
_  O  X
X  O  X
Jogador 1 vence!
```

2) Programa: sequencia.cpp (4,0 pontos)

Ler uma sequência **s1** com **n** inteiros no intervalo de [1, 100] e depois outra sequência **s2** com **k** inteiros também no intervalo de [1, 100]. Em seguida, verificar se a sequência **s2** está contida na sequência **s1** (na mesma ordem que aparece em **s2**). A sequência **s2** pode aparecer em **s1** de frente para trás ou de trás para frente. Ao final, imprimir a posição de **s1** na qual **s2** começa e o sentido, ou imprimir "não encontrado" se **s2** não ocorre em **s1**.

Exemplo 1:
```
s1 = [12, 3, 45, 9, 27, 4, 8, 58, 19, 72, 84, 36]
s2 = [27, 4, 8, 58]
s2 ocorre em s1 na posição 4 de frente para trás.
```

Exemplo 2:
```
s1 = [12, 3, 45, 9, 27, 4, 8, 58, 19, 72, 84, 36]
s2 = [72, 19, 58, 8, 4]
s2 ocorre em s1 na posição 9 de trás para frente.
```

Exemplo 3:
```
s1 = [12, 3, 45, 9, 27, 4, 8, 58, 19, 72, 84, 36]
s2 = [72, 84, 36, 10]
s2 não encontrado em s1.
```

Exemplo 4:
```
s1 = [12, 3, 45, 9, 27]
s2 = [72, 84, 36, 10, 7, 22]
s2 não encontrado em s1.
```

Restrições:

a) A relação entre **n** e **k** pode ser qualquer uma (n < k, n = k ou n > k).

b) n > 1 e k > 1.

c) As sequências **s1** e **s2** só podem conter inteiros no intervalo [1, 100]. Se o usuário digitar um valor fora desse intervalo, esse deve ser desconsiderado.

d) O primeiro elemento da sequência está na posição 0 (zero).

e) Todo o código deverá estar implementado na função main, sem o uso de funções auxiliares.