

## Exercício Prático - CI 208 - Programação de Computadores

O objetivo do exercício é implementar o jogo "Liga 4".

### O Jogo

O jogo "Liga 4" é um jogo para dois jogadores, formado por um tabuleiro de N linhas por M colunas. Cada jogador é representado por um determinado caracter ('O' para o jogador 1, e 'X' para o jogador 2). O objetivo do jogo é que cada jogador consiga arranjar seus caracteres em uma sequência de 4. Linhas horizontais, verticais ou diagonais são válidas. O jogador pode apenas escolher uma coluna para posicionar seu caracter sendo ela então colocada na sua primeira posição livre (de baixo para cima). Ou seja, caso exista um ou mais caracteres na coluna em que ele escolher, o caracter só poderá ser posto na posição livre subsequente. Se não houver espaço nesta coluna, ela não pode ser escolhida pelo jogador. As jogadas são feitas de maneira alternada entre o jogador um e o jogador dois. Ao mesmo tempo que um jogador tenta alcançar seu objetivo, ele pode atrapalhar as jogadas do seu oponente. Vence aquele que primeiro alcançar o objetivo de colocar seus 4 caracteres em sequência.

### Implementação

Assumiremos que o jogador 1 começa a partida. O jogo consiste basicamente nos seguintes passos:

- Mostrar uma representação do tabuleiro na tela;
- Pedir ao jogador 1 a coluna em que ele quer jogar sua peça. Se o jogador escolher uma coluna inválida (cheia) o programa deve emitir uma mensagem de erro correspondente e pedir novamente a jogada até que ela seja válida;
- Repetir o processo para o segundo jogador e assim sucessivamente.

O programa deve ser encerrado ao encontrar um vencedor ou até que não existam mais posições livres no tabuleiro, indicando um empate. O trabalho deverá ser implementado utilizando **necessariamente** um conjunto de funções especificadas a seguir. A saída gerada pelo programa, assim como sua entrada de dados, devem também obedecer rigorosamente o padrão estabelecido nesta especificação.

### Interface

A saída do programa consiste de:

- Representação do tabuleiro;
  - Mensagem indicando a vez de qual jogador (1 ou 2);
  - Mensagem indicando se uma jogada é inválida;
  - Mensagem indicando um vencedor.
- A impressão de um tabuleiro vazio deverá ser representada da seguinte maneira:

```
1 2 3 4 5 6 7
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Um tabuleiro preenchido após quatro rodadas poderia ter o seguinte layout visual:

```
1 2 3 4 5 6 7
. . . . .
. . . . .
. . . . .
. . X . . .
. . 0 X 0 . .
```

A cada rodada, o programa deve indicar de qual jogador é a vez, para que ele forneça o valor (inteiro) de uma coluna. A seguinte mensagem deverá ser gerada como saída, caso seja a vez do jogador 1:

1>

Ou então, caso seja a vez do jogador 2:

2>

No caso do jogador fornecer uma coluna inválida (fora da faixa válida) ou uma coluna já totalmente preenchida, o programa deve mostrar a seguinte mensagem:

Jogada inválida.

Lembrando que a única entrada do programa é a coluna escolhida pelo jogador a cada rodada. Ao encontrar um vencedor, o programa deve mostrar a seguinte mensagem antes de ser encerrado, caso o jogador 1 tenha ganho a partida:

Fim! 1

Ou então, caso o jogador 2 tenha ganhado:

Fim! 2

Ou ainda, no caso que não haja vencedor:

Empate!

### Especificação das Funções

O trabalho terá que obrigatoriamente utilizar funções na solução do problema. Todas as funções devem trabalhar com um tabuleiro cujas dimensões são definidas pelas seguintes constantes:

```
#define N 6  
#define M 7
```

Sendo N o número de linhas e M o de colunas. O programa **deve** funcionar se for escolhido para N e M qualquer outro valor inteiro positivo. As seguintes funções devem ser implementadas:

```
void escreveTabuleiro(int tab[N][M]);
```

- Mostra o tabuleiro *tab* na tela.

```
int ganhou(int tab[N][M], int jogador);
```

- Verifica se o *jogador* no tabuleiro *tab* ganhou. Devolve 1 se o jogador ganhou, devolve 0 caso contrário.

```
int estaCheio(int tab[N][M]);
```

- Verifica se o tabuleiro *tab* esta cheio. Devolve 1 se verdadeiro ou 0 caso contrário.

```
int leJogada(int jogador);
```

- Recebe como parâmetro *jogador* que pode ser 1 ou 2, lê a jogada e devolve a coluna escolhida.

```
void executaJogada(int tab[N][M], int jogador, int coluna);
```

- Executa a jogada indicada por *coluna* do jogador *jogador* no tabuleiro *tab*. (Preenche a posição correspondente da matriz).

```
int jogadaValida(int tab[N][M],int coluna);
```

- A partir de *coluna* escolhida pelo jogador, devolve 1 caso a jogada seja válida no tabuleiro *tab* ou devolve 0 caso seja inválida.