

## Regras para a execução do trabalho:

1. No dia **13/07/2018** às **23:50** encerra-se o prazo para a entrega do trabalho prático II. Trabalho atrasados serão descontados em 0,1 pontos (da média final) por dia de atraso.
2. O trabalho será realizado **individualmente**.
3. O aluno, até o prazo final de entrega, deverá fazer o envio **via Moodle** dos arquivos de código-fonte do programa e as instruções de como compilar e/ou executar.
4. O arquivo deve possuir um cabeçalho contendo o nome completo do aluno e respectivo número de matrícula, além de uma breve descrição da solução do programa. O código deve estar indentado e devidamente comentado para facilitar o entendimento. Códigos mal identados e/ou não comentados não serão considerados (**nota zero**).

## Sobre a avaliação do trabalho:

1. O peso do trabalho corresponde a 3 pontos na média final.
2. A nota será composta por dois itens de avaliação, descritos a seguir:
  - a) **Qualidade das soluções geradas:** a nota será de acordo com a estratégia proposta (até 2.5 pontos).
  - b) **Apresentação da solução:** A solução desenvolvida será defendida/apresentada em dia/hora definidos *a priori*. (0.5 ponto).
3. Trabalhos plagiados, com informações parciais e/ou não devidamente apresentados terão notas **zero**.

## Datas:

1. **Apresentação do enunciado do trabalho:** 23/06/2018
2. **Entrega do trabalho:** 13/07/2018 até às 23:50 (via Moodle).

## Definição do problema:

Tic-tac-toe (em português, “jogo-da-velha”) é um exemplo de um  $m, n, k$  – game, onde dois jogadores revezam as jogadas em um tabuleiro  $m \times n$  – até que um deles obtenha  $k$  elementos (“O” ou “X”) em uma fileira. O trabalho II de IA consiste implementar a inteligência de um BOT para jogar o Tic-tac-toe contra um “humano” ou contra um outro BOT. O BOT deve ser capaz de jogar em qualquer tabuleiro  $m \times n$  e para qualquer valor de  $k$ . Observa na Figura 1 um exemplo de um Tic-tac-toe com dimensões  $5 \times 5$  e com  $k = 4$ .

O	X	X	O	O	X
X	O	X	O	O	X
O	X	O	X	X	O
O	X	O	X	O	X
X	O	X	O	X	O
X	O	O	X	X	O

Figura 1: Exemplo de um jogo-da-velha com dimensões  $5 \times 5$  (i.e.,  $m$  e  $n$ ) e  $k = 4$ . Neste caso, o “X” ganha o jogo.

## Entrada:

O algoritmo desenvolvido deverá receber como entrada as dimensões  $m$ ,  $n$ ,  $j$  (o jogador), a constante  $k$  e um estado do jogo Tic-tac-toe. O jogador “X” é representado pelo inteiro 1, o jogador “O” pelo inteiro 2 e posições em branco no tabuleiro pelo inteiro 0. Exemplo de possíveis entradas validas incluem:

Exemplo de uma entrada  $3 \times 3$ ,  $k = 3$  sendo jogado por “X”.

```
./a.out 3 3 1 3 0 0 0 0 0 0 0 0 0
```

Exemplo de uma entrada  $3 \times 3$ ,  $k = 3$  sendo jogado por “O”.

```
./a.out 3 3 2 3 1 1 2 0 0 0 2 1 0
```

Exemplo de uma entrada  $4 \times 4$ ,  $k = 2$  sendo jogado por “O”.

```
./a.out 4 4 2 2 1 2 1 2 0 0 0 0 0 0 0 0 0 0 1 0
```

## Saída:

O algoritmo desenvolvido deverá imprimir na saída padrão do sistema operacional o resultado da jogado pelo BOT através de um par de índices  $ij$ , sendo ( $i < m$  e  $j < n$ ).

Exemplo de uma saída válida:

```
1 1
```

Exemplo de uma saída válida:

```
2 1
```