Considere o problema do Cavalo, ou *Knight's tour* (https://en.wikipedia.org/wiki/Knight%27s_tour). Pretende-se que implemente uma solução para um tabuleiro irregular N x M, com casas não utilizáveis, e que o objetivo seja fazer um percurso sem repetição de casas, com uma casa inicial, e tenha comprimento K, não obrigatoriamente igual ao número de casas livres, e aberto, ou seja, não tem de voltar à casa inicial. O movimento do cavalo segue as regras do xadrez: move-se em L (duas casas para a esquerda/direita, e uma casa para cima/baixo, ou então uma casa para a esquerda/direita, e duas casas para cima/baixo).

Exemplo de um problema:

```
P1
.#####
##.@.#
#....#
#...##
#####.
9
```

Este problema tem identificação 1 (linha P1) e tem dimensão 6 x 5, dado que tem 6 colunas e 5 linhas, estando as casas não visitáveis com o símbolo **#**, as casas livres com o símbolo **.** e a casa com a posição inicial do cavalo, com o símbolo **@**. Todas as linhas têm o mesmo número de caracteres, e o número de casas livres é neste caso 11 enquanto que o comprimento necessário do caminho a encontrar é 9.

O problema poderia ser resolvido pela seguinte sequência de movimentos válidos:

```
.#####
##.@.#
#....#
#...##
#####.
(1)
.#####
##.#.#
#....#
#.@.##
#####.
(2)
.#####
##.#.#
#...@#
#.#.##
#####.
(3)
```

```
.#####
##.#.#
#...##
#.#.##
#####@
(4)
.#####
##.#.#
#...##
#.#@##
######
(5)
.#####
##.#.#
#@..##
#.####
######
(6)
```

```
@#####
##.#.#
##..##
#.####
######
(7)
######
##@#.#
##..##
#.####
######
(8)
######
####.#
##..##
#@####
######
(9)
######
####.#
##.@##
######
######
```

Os testes devem ser realizados com base no ficheiro **puzzles.txt**, que contém problemas nestas condições, de aumento de complexidade gradual. Pretende-se que possa ser carregado qualquer um dos problemas, devendo o programa aceitar os seguintes argumentos:

C:\...>efolioA <ficheiro> <problema>

e-fólio A

O número do problema é 1 para o primeiro problema, 2 para o segundo, e assim sucessivamente, iniciando-se com a linha P1, P2, etc. No caso do ficheiro fornecido existem 20 problemas, sendo este o tipo de problema que se pretende resolver com apenas técnicas de procuras cegas.

Deve entregar:

- Relatório;
- Código fonte dos algoritmos implementados.

Critérios de correção (4 valores):

- **Análise do problema** (2 valores): Referência a aspectos importantes do problema no relatório, revelando independentemente de os implementar ou não, que tinha consciência dos mesmos.
- **Identificação de algoritmos** (1 valor): Identificação clara dos algoritmos que implementou de acordo com a nomenclatura do livro e da UC, juntamente com as configurações utilizadas, ou no caso de utilização de um algoritmo distinto, deve descrevê-lo. A utilização de outro nome para os mesmos algoritmos é possível, desde que indique a qual correspondente. A penalização para a não identificação corresponde a 0,5 valores.
- **Resultados** (1 valor): Deve procurar resolver pelo menos um problema, e caso consiga resolver vários, deve procurar resolver o problema de nível mais elevado (deve sempre utilizar menos de 1 minuto de CPU, modo de *release*), e indicar o algoritmo/configuração, tempo utilizado, número de expansões e gerações.

O trabalho é individual mas caso os estudantes pretendam, podem partilhar resultados. A partilha de resultados afasta o problema de uma situação real, em que não existindo referências, não se sabe até onde se consegue chegar, mas pode contribuir para uma maior participação no e-folio, e em nada afeta a avaliação. Os resultados obtidos através da resolução de exemplo, serão conhecidos após o lançamento das notas.