

MC102 - Algoritmos e Programação de Computadores**Turmas QRSTWY****Instituto de Computação - Unicamp****Professores:** Hélio Pedrini e Zanoni Dias**Monitores:** Andre Rodrigues Oliveira, Gustavo Rodrigues Galvão, Javier Alvaro Vargas Muñoz e Thierry Pinheiro Moreira

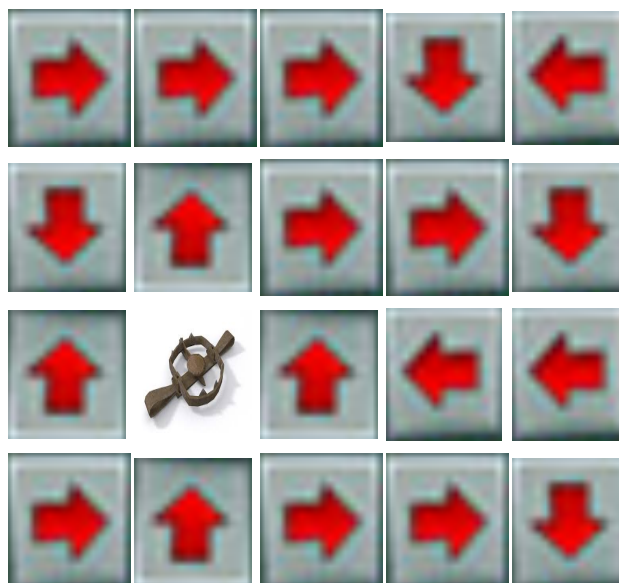
Lab 07b - Walk-Bot - Part I

Prazo de entrega: 04/05/2015 às 13h59m59s**Peso:** 5

Vendo o sucesso do Cargo-Bot, a empresa LastLife resolveu lançar um jogo concorrente, chamado Walk-Bot. Na verdade, eles ainda estão na fase de prototipagem e testes com usuários. Nesta primeira versão do jogo, um robô (ou Walk-Bot) deve andar por um tabuleiro tal que, em cada casa, há uma instrução sobre qual deve ser o próximo passo do robô: prosseguir em uma determinada direção ou parar (no caso, o robô para quando entra numa casa correspondente a uma armadilha, que o destrói). A ideia básica do jogo é que o robô deve atravessar o tabuleiro, isto é, começar em uma das casas localizadas na coluna mais à esquerda do tabuleiro e terminar em uma das casas localizadas na coluna mais à direita do tabuleiro, na qual deve existir uma instrução para ele ir para direita. A casa inicial deve ser escolhida pelo jogador.

O desafio do jogo está em não só conseguir escolher uma casa inicial que leve o Walk-Bot a atravessar o tabuleiro, mas também escolher uma casa inicial que o leve a fazê-lo com o menor número de passos. Isso porque, a cada passo, o robô perde uma quantidade de energia, que é parcialmente carregada a cada mudança de fase (ou tabuleiro). Desse modo, se o caminho de travessia for muito longo, a energia do Walk-Bot acaba e ele morre. Além disso, conforme as fases vão avançando, maiores são os tabuleiros e, portanto, mais energia é necessária para atravessá-los.

Para entender melhor como o jogo funciona, considere que o Walk-Bot deve atravessar o tabuleiro ilustrado abaixo.





Dado que uma casa localizada na linha i e na coluna j pode ser identificada pelo par (i, j) e que a casa mais acima e mais à esquerda do tabuleiro é a casa $(1, 1)$, nós podemos traçar os seguintes caminhos para o Walk-Bot no tabuleiro ilustrado acima:

- $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (1, 4) \rightarrow (2, 4) \rightarrow (2, 5) \rightarrow (3, 5) \rightarrow (3, 4) \rightarrow (3, 3) \rightarrow (2, 3) \rightarrow (2, 4) \dots$
- $(2, 1) \rightarrow (3, 1) \rightarrow (2, 1) \dots$
- $(3, 1) \rightarrow (2, 1) \rightarrow (3, 1) \dots$
- $(4, 1) \rightarrow (4, 2) \rightarrow (3, 2)$
- $(5, 1) \rightarrow (5, 2) \rightarrow (5, 3) \rightarrow (4, 3) \rightarrow (4, 4) \rightarrow (4, 5) \rightarrow (5, 5)$

Note que os três primeiros caminhos terminam com "...". Isso significa que o Walk-Bot entrou em *loop* e, conseqüentemente, morrerá em algum momento devido ao término de sua energia. Note também que o quarto caminho leva a uma armadilha. Portanto, nesse caso, o único caminho que leva o Walk-Bot a atravessar o tabuleiro é aquele que começa na casa $(5, 1)$. Tenha em mente que este foi apenas um exemplo. Podem existir tabuleiros em que há mais de um caminho de travessia. Além disso, você pode considerar que o Walk-Bot só consegue sair do tabuleiro pelas casas localizadas na coluna mais à direita. Em outras palavras, as casas localizadas na primeira linha (*i.e.* na linha mais acima) nunca terão uma instrução para o Walk-Bot ir para cima, as casas localizadas na última linha (*i.e.* na linha mais abaixo) nunca terão uma instrução para o Walk-Bot ir para baixo e, finalmente, as casas localizadas na primeira coluna (*i.e.* na coluna mais à esquerda) nunca terão uma instrução para o Walk-Bot ir para esquerda.

Você foi um dos usuários selecionados para testar o jogo e dar um *feedback* para a LastLife. A fim de dar um bom *feedback*, você decidiu escrever um programa que, dado um tabuleiro, faz um relatório de todos os caminhos possíveis a serem percorridos pelo Walk-Bot. Com esse relatório, é possível classificar a dificuldade de cada tabuleiro. Mais especificamente, para cada casa inicial possível de ser escolhida, seu relatório deverá indicar se:

- o Walk-Bot consegue atravessar o tabuleiro. Neste caso, seu programa deverá indicar o número de passos dados até a última casa do tabuleiro visitada pelo Walk-Bot. No exemplo anterior, nós temos que o Walk-Bot deu 6 passos para atravessar o tabuleiro, isto é, para caminhar da casa $(5, 1)$ até a casa $(5, 5)$;
- o Walk-Bot cai em uma armadilha. Neste caso, seu programa deverá indicar o número de passos dados até o Walk-Bot cair na armadilha. No exemplo anterior, o Walk-Bot deu 2 passos até cair na armadilha, isto é, da casa $(4, 1)$ até a casa $(3, 2)$;
- o Walk-Bot entra em um *loop*. Neste caso, como não faz sentido listar o número de passos dados, seu programa deverá indicar o tamanho do *loop*, que é igual ao número de casas (ou posições) distintas que o constituem. No exemplo anterior, nós temos que quando o Walk-Bot inicia a travessia da casa $(1, 1)$, ele fica preso em um *loop* de tamanho 6, formado pelas casas $(2, 4)$, $(2, 5)$, $(3, 5)$, $(3, 4)$, $(3, 3)$ e $(2, 3)$. Similarmente, quando o Walk-Bot inicia a travessia da casa $(2, 1)$ ou da casa $(3, 1)$, ele fica preso em um *loop* de tamanho 2, formado pelas casas $(2, 1)$ e $(3, 1)$.






Entrada

A entrada é constituída de várias linhas, tal que:

- A primeira linha da entrada contém 2 números inteiros, N e M , onde N representa o número de

linhas do tabuleiro, com $5 \leq N \leq 50$, e M representa o número de colunas do tabuleiro, com $5 \leq M \leq 50$;

- As próximas N linhas contêm M caracteres cada, pertencentes ao conjunto $\{L, R, D, U, X\}$. Este conjunto possui uma correspondência de um-para-um com as possíveis casas de um tabuleiro, de tal modo que:

- $L =$  ;
- $R =$  ;
- $D =$  ;
- $U =$  ;
- $X =$  .

Saída

A saída é constituída de N linhas, de tal modo que a linha i indica qual o resultado de se iniciar a travessia na casa $(i, 1)$. Mais especificamente, nós temos que:

- caso o Walk-Bot saia do tabuleiro, deverá ser impressa uma linha no formato "Saiu em (i, M) apos X passo(s)", onde (i, M) indica a última casa em que o Walk-Bot esteve antes de sair do tabuleiro e X indica o número de passos dados no percurso (vide explicação no enunciado);
- caso o Walk-Bot caia em uma armadilha, deverá ser impressa uma linha no formato "Armadilha em (i, j) apos Y passo(s)", onde (i, j) indica a casa em que se encontra a armadilha e Y indica o número de passos dados no percurso (vide explicação no enunciado);
- caso o Walk-Bot entre em *loop*, deverá ser impressa uma linha no formato "Loop de tamanho Z ", onde Z indica o tamanho do *loop*. O tamanho de um *loop* é dado pelo número de casas (ou posições) distintas que o constituem (vide explicação no enunciado).

Exemplos

#	Entrada	Saída
1	5 5 RRRDL DURRD UXULL RURRD RRUXR	Loop de tamanho 6 Loop de tamanho 2 Loop de tamanho 2 Armadilha em (3,2) apos 2 passo(s) Saiu em (5,5) apos 6 passo(s)
2	6 6 RRRRRD RRRRUL RRRRRD RRRRRD RXRRRR XRRRRR	Loop de tamanho 4 Loop de tamanho 4 Saiu em (5,6) apos 7 passo(s) Saiu em (5,6) apos 6 passo(s) Armadilha em (5,2) apos 1 passo(s) Armadilha em (6,1) apos 0 passo(s)
3	5 7 DRDRDRD DUDUDUD DUDUDUD DUDUDUD RURURUR	Saiu em (5,7) apos 34 passo(s) Saiu em (5,7) apos 33 passo(s) Saiu em (5,7) apos 32 passo(s) Saiu em (5,7) apos 31 passo(s) Saiu em (5,7) apos 30 passo(s)
4	9 8 RRRRRRRD	Armadilha em (4,5) apos 27 passo(s) Armadilha em (4,5) apos 12 passo(s)

	DLLLLLLL RRRRRRRD RRRDLLLL RRDRRRRD RRRRRRDD ULLLLLLD RRRRRRUR RRRRRRRX	Armadilha em (4,5) apos 11 passo(s) Saiu em (8,8) apos 11 passo(s) Loop de tamanho 14 Loop de tamanho 14 Loop de tamanho 14 Loop de tamanho 14 Armadilha em (9,8) apos 7 passo(s)
5	5 10 RRRRRRRRRD UDLLLLLLLL URRRRRRRRD ULLLLLLLLLL RRRRRRRRRR	Loop de tamanho 40 Loop de tamanho 40 Loop de tamanho 40 Loop de tamanho 40 Saiu em (5,10) apos 9 passo(s)