

Caça ao Tesouro

Exemplo

Mini-Projeto 1

ATAD 2016

Mini-Projeto 1

ATAD 2016

[illegible]

A 10x10 grid with columns labeled 0 through 9 and rows labeled 0 through 9. A blue arrow points to the cell at (0,0), which contains a small blue flag icon.

Ficheiros de Entrada (1)

“mapa.txt”

10 10

(1,1)

(1,9)

(3,8)

(4,1)

(5,6)

(7,0)

(7,7)

(9,4)

(9,8)










Dimensão do mapa
n linhas x m colunas

Localizações dos Tesouros
(linha,coluna)

Localizações dos Tesouros
(linha,coluna)

Ficheiros de Entrada (2)

“mapa.txt”

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

10 10
(1,1)
(1,9)
(3,8)
(4,1)
(5,6)
(7,0)
(7,7)
(9,4)
(9,8)

10 10
(1,1)
(1,9)
(3,8)
(4,1)
(5,6)
(7,0)
(7,7)
(9,4)
(9,8)

Ficheiros de Entrada (3)

“movimentos.txt”

MOVE(2)	LOOP(3,2)
LOOP(3,3)	LOOP(1,2)
JUMP(5,4)	MOVE(4)
MOVE(3)	LOOP(2,6)
LOOP(1,3)	MOVE(4)
MOVE(3)	MOVE(4)
MOVE(1)	LOOP(4,3)
UNDO(3)	JUMP(8,2)
MOVE(3)	MOVE(2)
MOVE(3)	LOOP(3,2)
LOOP(1,2)	LOOP(1,3)

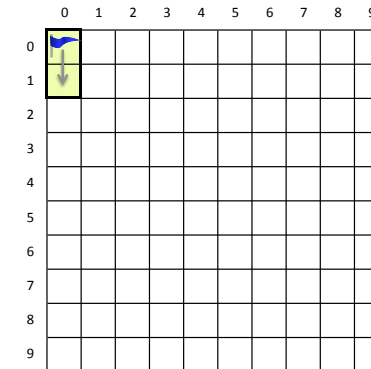


Movimentos (1)

MOVE

MOVE(d) : avançar para a localização imediatamente a **d** da localização atual, onde **d** pode ser **1** (Norte), **2** (Sul), **3** (Este) ou **4** (Oeste).

MOVE(2)

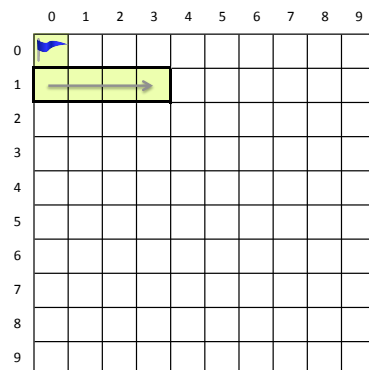


Movimentos (2)

LOOP

LOOP(d, n) : avançar **n** vezes na direção **d**, onde **d** pode ser **1** (Norte), **2** (Sul), **3** (Este) ou **4** (Oeste).

LOOP(3,3)



Movimentos (3)

JUMP

JUMP(x,y) : avançar para a localização **(x, y)**, percorrendo o caminho mais curto (se existirem vários, fica ao critério dos alunos escolher um).

Observação:

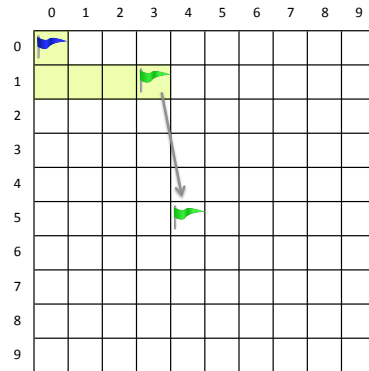
O movimento “JUMP” pode representar uma linha diagonal entre posições da matriz, para a qual não existe um caminho definido.

Há que calcular (a forma de cálculo fica ao critério dos alunos) que localizações são visitadas aquando de um movimento “JUMP” para ser possível construir o percurso.

Movimentos (4)

JUMP

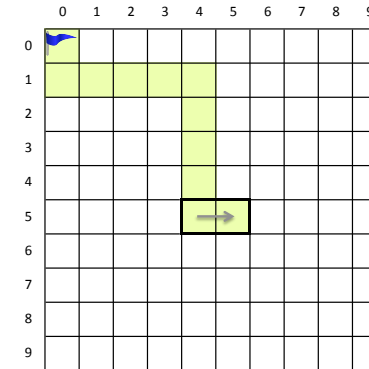
JUMP(5,4)



Movimentos (5)

...

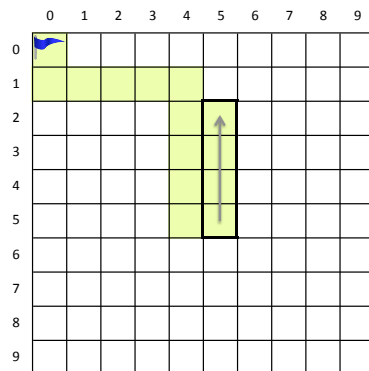
MOVE(3)
 LOOP(1,3)
 MOVE(3)
 MOVE(1)
 UNDO(3)



Movimentos (6)

...

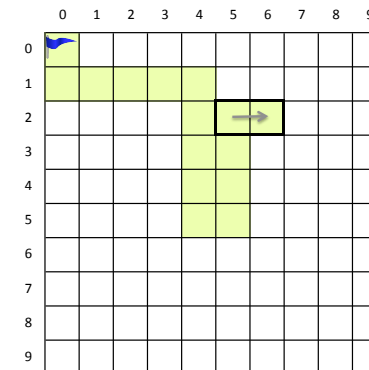
MOVE(3)
LOOP(1,3)
 MOVE(3)
 MOVE(1)
 UNDO(3)



Movimentos (7)

...

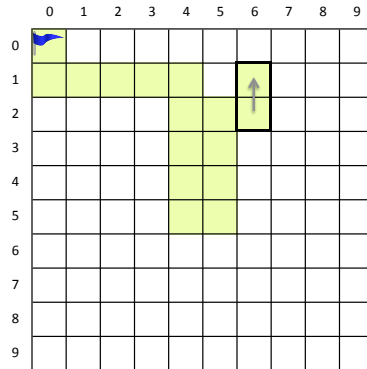
MOVE(3)
 LOOP(1,3)
MOVE(3)
 MOVE(1)
 UNDO(3)



Movimentos (8)

...

MOVE(3)
LOOP(1,3)
MOVE(3)
MOVE(1)
UNDO(3)



Movimentos (9)

UNDO

UNDO(n) (onde $n \geq 1$) : anular os n movimentos “MOVE” e LOOP” imediatamente anteriores e consecutivos, isto é, não existem movimentos “JUMP” ou “UNDO” entre eles.

MOVE(3)
LOOP(1,3)
MOVE(3)
MOVE(1)
UNDO(3)

Há que anular os últimos 3 movimentos:

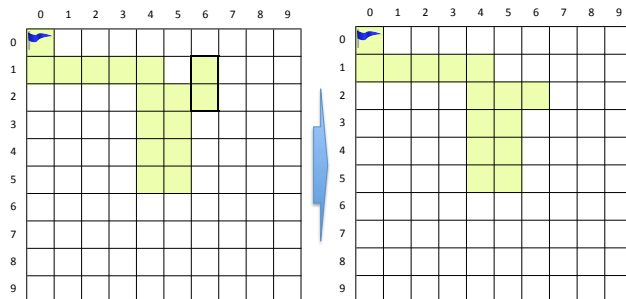
LOOP(1,3)
MOVE(3)
MOVE(1)

Movimentos (10)

UNDO

UNDO(n) (onde $n \geq 1$) : anular os n movimentos “MOVE” e LOOP” imediatamente anteriores e consecutivos, isto é, não existem movimentos “JUMP” ou “UNDO” entre eles.

LOOP(1,3)
MOVE(3)
MOVE(1)
UNDO(3)

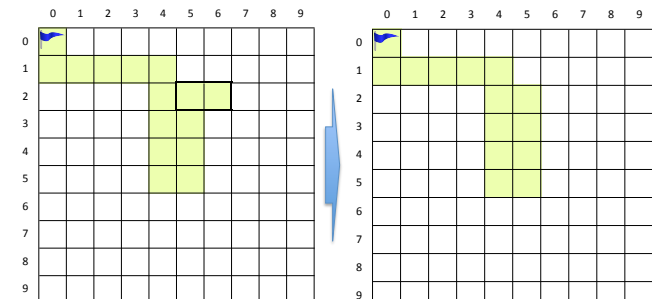


Movimentos (11)

UNDO

UNDO(n) (onde $n \geq 1$) : anular os n movimentos “MOVE” e LOOP” imediatamente anteriores e consecutivos, isto é, não existem movimentos “JUMP” ou “UNDO” entre eles.

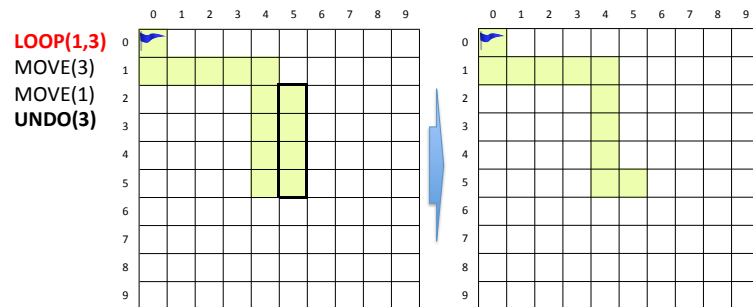
LOOP(1,3)
MOVE(3)
MOVE(1)
UNDO(3)



Movimentos (12)

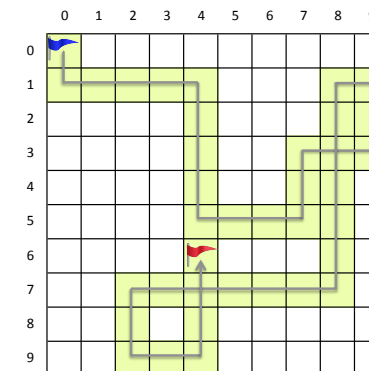
UNDO

UNDO(n) (onde $n \geq 1$) : anular os n movimentos “MOVE” e LOOP” imediatamente anteriores e consecutivos, isto é, não existem movimentos “JUMP” ou “UNDO” entre eles.

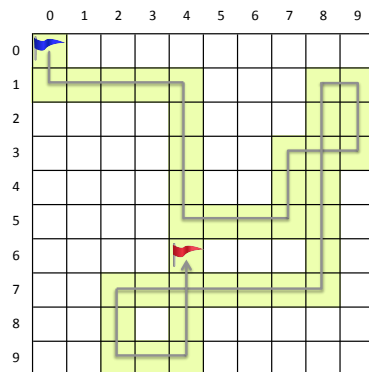


Movimentos (13)

... Fim



Percurso



(0,0)	(4,7)	(7,7)
(1,0)	(3,7)	(7,6)
(1,1)	(3,8)	(7,5)
(1,2)	(3,9)	(7,4)
(1,3)	(2,9)	(7,3)
(2,3)	(1,9)	(8,3)
(3,3)	(1,8)	(8,2)
(4,3)	(2,8)	(9,2)
(5,3)	(3,8)	(9,3)
(5,4)	(4,8)	(9,4)
(5,5)	(5,8)	(8,4)
(5,6)	(6,8)	(7,4)
(5,7)	(7,8)	(6,4)

Ficheiro de Saída (1)

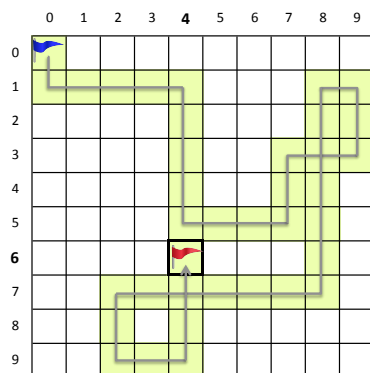
“resultados.txt”

(6,4) ← Localização do Final do Percurso
 39 ← Distância Percorrida
 6 ← Número de Tesouros encontrados
 (7, 4, 2) } Localizações do percurso
 (3, 8, 2) } por onde houve mais de uma passagem
 e número de passagens

Ficheiro de Saída (2)

Localização do Final do Percurso

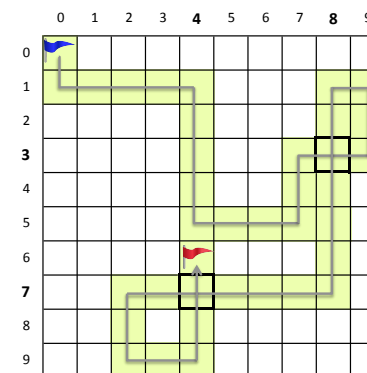
(6,4)
39
6
(7, 4, 2)
(3, 8, 2)



Ficheiro de Saída (3)

Localizações Repetidas

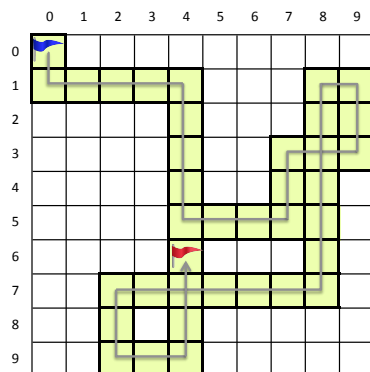
(6,4)
39
6
(7, 4, 2)
(3, 8, 2)



Ficheiro de Saída (4)

Distância Percorrida

(6,4)
39
6
(7, 4, 2)
(3, 8, 2)

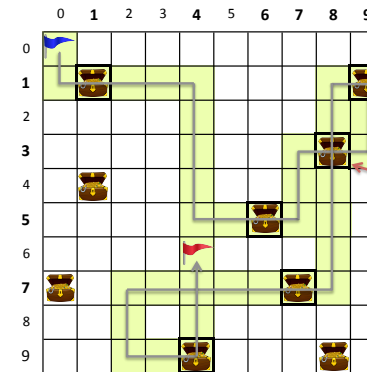


Distância Percorrida = Número de Localizações Visitadas, incluindo repetições.

Ficheiro de Saída (5)

Número de Tesouros Encontrados

(6,4)
39
6
(7, 4, 2)
(3, 8, 2)



Cada Tesouro só é encontrado uma vez!

Boa Caça ao Tesouro!