

Exercício de Programação 3 – Recursividade

Valor: 5 pontos

Data de devolução: 26/09/2017

O Problema

Neste exercício de programação deverá ser criado um programa recursivo que encontre o caminho em um labirinto que leve um rato de sua posição inicial até um pedaço de queijo.

A configuração inicial do labirinto deverá ser lida de um arquivo de texto com um conteúdo como o abaixo:

=====	=====

	. . .
=	= . . .
	*** .
=====	* * =====
	* *****
=====	* ===== *
	***** *
= =	= = * *
	*** *
= =	= * = *
M C	***** *****
=====	=====

(a)

(b)

Figure 1: Formato de descrição dos labirintos: (a) configuração inicial de um labirinto, (b) uma solução possível para o problema.

Todo arquivo terá sempre 20 linhas e 20 caracteres por linha (além do ‘\n’ ao final de cada linha). Os caracteres usados para descrever o estado inicial do labirinto são: “|”, “=”, “ ” (espaço em branco), “M”, e “C”. Os caracteres “|” e “=” representam paredes, “ ” representa uma posição livre, “M” a posição inicial do rato, e “C” a posição do queijo.

O objetivo do exercício é encontrar um caminho entre as posições “M” e “C”, que não atravesse nenhuma parede e que não passe duas vezes pela mesma posição.

Uma solução possível para o labirinto da Figura 1a pode ser vista na Figura 1b, onde o caminho da solução está marcado pelos caracteres “*”. Os caminhos que foram percorridos durante a busca, mas que entretanto não levaram a uma solução, estão marcados pelos caracteres “.”.

Observe que a solução vista na figura 1b não é a única possível. Dependendo da ordem em que os caminhos são explorados em cada encruzilhada outra configuração final é possível.

Requerimentos de código

Todo o código desenvolvido deverá ser escrito na linguagem C, e somente poderá ser utilizada a biblioteca padrão da linguagem.

A resolução do problema deverá ser desenvolvida utilizando obrigatoriamente chamadas de funções recursivas, e a solução do problema ao final da execução deverá ser impressa na tela, e estar no mesmo formato da solução vista na Figura 1b. O caminho que leva da posição inicial à posição final deverá ser demarcado pelo caractere “*”, os caminhos que foram explorados mas que não levaram a uma solução deverão ser demarcados pelo caractere “.”, e os caminhos que não chegaram a ser explorados deverão continuar demarcados com o caractere “ ” (espaço em branco).

Apesar de não ser a única forma possível de se estruturar um programa que resolva o exercício, sugere-se que a função principal tenha uma assinatura como:

```
int resolve_labirinto(Labirinto labirinto, int i, int j);
```

O tipo `Labirinto` é um array bidimensional de `char`, capaz de representar o maior labirinto possível:

```
typedef char Labirinto[20][20];
```

e os argumentos `int i` e `int j` determinam a linha e coluna da posição a ser analisada. O valor inteiro retornado informa se uma solução foi encontrada passando pela posição atual (1), ou se uma parede ou uma posição já explorada foi encontrada (0).

Basicamente, a sua função recursiva deverá verificar a posição (i,j) do labirinto. Se for uma parede ou um caminho já explorado, deverá retornar 0. Se achar o queijo, retornar 1. Senão deve investigar recursivamente cada um dos vizinhos (cima, baixo, direita e esquerda), retornando o resultado dessa investigação. A função também deverá marcar o tabuleiro de acordo (. ou *). A função em si não é complicada, mas você deverá entender os conceitos de recursividade para implementá-la corretamente.

O que deve ser entregue

Todo o código fonte do programa desenvolvido (arquivos .c e .h).

Comentários gerais

1. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
2. Clareza, indentação e comentários no programa são fundamentais.
3. Não utilize bibliotecas específicas de algum sistema operacional e que não façam parte da distribuição padrão da linguagem C.
4. O trabalho é individual
5. Trabalhos copiados serão penalizados conforme anunciado na primeira aula.
6. Penalização por atraso: 1 dia 50% da nota. 2 dias: 100% da nota.