

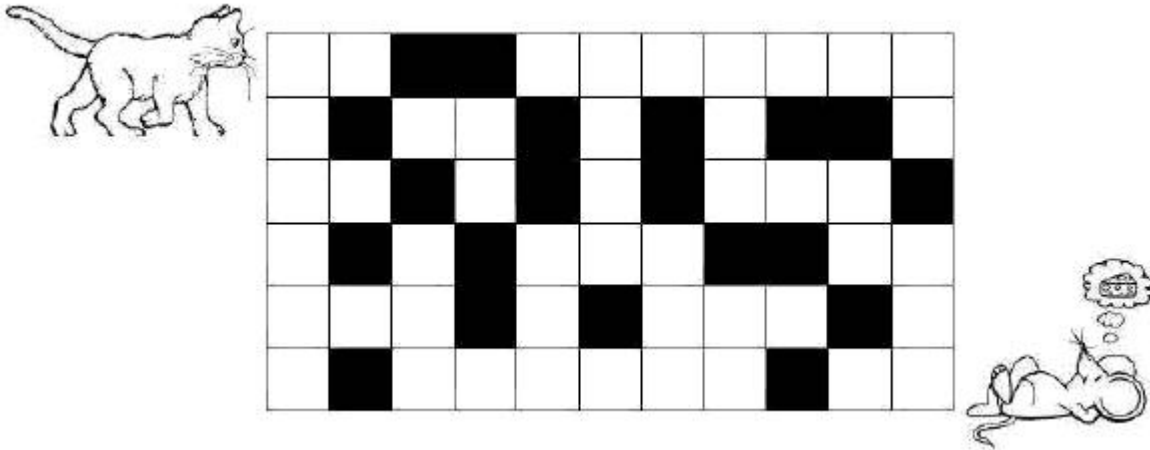
## TERCEIRO EXERCÍCIO PROGRAMA

### MAC0110 - BMAC - NOTURNO

Professor: Roberto Cesar

## 1 O Gato e o Rato

Considere um labirinto como o do desenho abaixo:



Um labirinto desses pode ser representado por uma matriz retangular  $L$ , cujo elemento  $L_{ij}$  vale 0 ou -1 conforme a casa correspondente do labirinto seja uma passagem livre ou uma parede, respectivamente.

Suponha agora que em alguma posição desse labirinto haja um gato, e também uma toca onde há um rato. A intenção é ajudar o gato a encontrar o rato.

Um método geral para resolver esse problema consiste em marcar com o número  $k$  ( $k = 1, 2, \dots$ ) as casas livres que estejam a exatamente  $k - 1$  passos de distância do rato, pelo caminho mais curto possível. Suponha que, a cada passo, o gato possa se deslocar de apenas uma casa na vertical ou na horizontal. Então, rotula-se inicialmente a posição da porta com 1 e para cada  $k \geq 2$  examinam-se todas as casas livres do labirinto, marcando-se com  $k$  aquelas ainda não marcadas e que sejam vizinhas a alguma casa marcada com  $k - 1$ .

A marcação continua até ser atingido um valor de  $k$  tal que nenhuma casa esteja em condições de ser marcada. Supondo que o labirinto seja  $L = [1..6, 1..11]$ , e que a toca do rato esteja na posição  $[6, 11]$ , ao final da marcação teremos a seguinte matriz:

26	27	-1	-1	12	11	10	9	10	11	12
25	-1	0	0	-1	12	-1	8	-1	-1	13
24	25	-1	0	-1	13	-1	7	6	5	-1
23	-1	21	-1	15	14	15	-1	-1	4	3
22	21	20	-1	16	-1	16	17	18	-1	2
23	-1	19	18	17	18	17	18	-1	2	1

Tendo feito esta marcação, um caminho mais curto do gato até o rato (se existir) pode ser determinado partindo-se da posição do gato e passando a cada etapa para uma casa vizinha

cuja numeração seja menor do que a atual. Por exemplo, se o gato estiver na posição  $[3, 2]$ , este precisará percorrer pelo menos 25 casas para chegar ao rato:

$[3, 2], [3, 1], [4, 1], [5, 1], [5, 2], [5, 3], \dots, [4, 10], [4, 11], [5, 11], [6, 11]$ .

## 2 Seu programa

Faça um programa em C para resolver o seguinte problema:

Dados dois inteiros positivos  $m, n$ , uma matriz  $L = [1..m, 1..n]$ , representando um labirinto, a posição  $(g1, g2)$  do gato e a posição  $(r1, r2)$  da toca do rato desse labirinto, determinar (se existir) um caminho mais curto que o gato deve percorrer até encontrar a toca onde está o rato. O labirinto é uma matriz de inteiros cujos elementos são 0 ou -1, conforme a posição correspondente do labirinto seja uma passagem livre ou uma parede, respectivamente.

Se tal caminho não existir, o programa deve imprimir uma mensagem dizendo que será impossível o gato achar o rato.

O seu programa deve fazer uso de (pelo menos) duas funções, como descrito abaixo.

- A função `void marque(int L[][MAXCOL], int m, int n, int r1, int r2);` que tem como parâmetros um labirinto, isto é, uma matriz de inteiros  $L = [1..m, 1..n]$ , suas dimensões  $m$  e  $n$ , e dois inteiros,  $r1$  e  $r2$ , que indicam a posição  $(r1, r2)$  da toca do rato nesse labirinto. Essa função deve efetuar a marcação da matriz  $L$ , como explicado anteriormente. (Após a chamada dessa função, o seu programa deve imprimir a matriz já marcada.)
- A função `void imprima_caminho(int L[][MAXCOL], int m, int n, int g1, int g2);` que tem como parâmetros uma matriz de inteiros  $L = [1..m, 1..n]$  já marcada, as dimensões  $m$  e  $n$  dessa matriz, e dois inteiros  $g1$  e  $g2$  indicando a posição  $(g1, g2)$  do gato nessa matriz. Esta função deve imprimir, caso exista, o caminho a ser percorrido pelo gato até o rato. Aqui, queremos a ordem em que as posições  $(i, j)$  da matriz devem ser percorridas. Adicionalmente, para uma visualização melhor desse caminho no labirinto, essa função deve imprimir também um outro labirinto (matriz tipo char), onde apareçam caracteres apropriados indicando as paredes, as posições livres, a posição do gato (por exemplo, a letra G), a posição do rato (por exemplo, a letra R), e o caminho que o gato deve seguir.

## 3 Entrada do Programa

O seu programa deverá ler os dados de entrada de um arquivo *labirinto.txt*. Tais dados já foram mencionados e são: as dimensões  $m$  e  $n$  do labirinto, a posição  $r1$  e  $r2$  do Rato, a posição  $g1$  e  $g2$  do Gato e o labirinto. Um exemplo de entrada está a seguir:

```
6 11
6 11
1 1
eepppeeeeeee
epeepepeppe
eepepepeeeep
epepeeepppep
eeepepeeepe
epeeeeeeepee
*
```

Tal entrada corresponde ao labirinto exemplo do enunciado. Note que na primeira linha estão  $m$  e  $n$ , na segunda  $r1$  e  $r2$ , na terceira  $g1$  e  $g2$ , o labirinto, onde `e` corresponde um espaço vazio e `p` uma parede, e por último o caracter `*` indica o fim do arquivo.

Existe um programa no paca que mostra como ler dados de um arquivo externo.

A função que lerá o arquivo de entrada deve ser definida da seguinte maneira: `void le_labirinto (int L[][MAXCOL], int *m, int *n, int *r1, int *r2, int *g1, int *g2);` que armazena as informações do arquivo na matriz e nas variáveis enviadas como parâmetros.

O SEU PROGRAMA DEVE SEGUIR O PADRÃO DO ARQUIVO EXEMPLO.

## 4 O que entregar

- Data de entrega: 02/07/2010;
- Entregar apenas um arquivo com o nome `ep3.c`;
- Para a entrega, utilize o PACA. Você poderá entregar várias versões de um mesmo EP até o término do prazo, mas somente a última versão que permanecerá armazenada pelo sistema;
- Caso o EP tenha sido feito em dupla, ambos os alunos devem enviar o arquivo no sistema PACA;
- Não serão aceitas submissões por e-mail ou atrasadas. Não deixe para a última hora, pois o sistema pode ficar congestionado, e você poderá não conseguir enviar;
- Guarde uma cópia do seu EP pelo menos até o final do semestre;
- No início do código, acrescente um cabeçalho conforme os EP1 e EP2.