

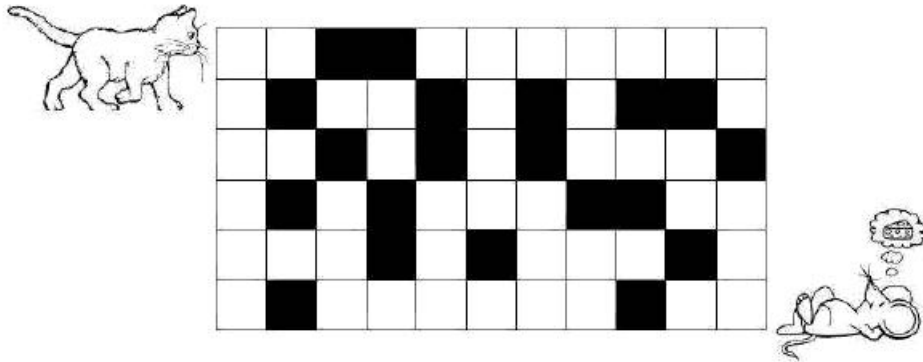
# Terceiro Exercício Programa

MAC 115 - Física Diurno 2010

Prof Roberto Marcondes Cesar Jr  
Entrega até 26/11/2010

## 1 O Gato e o Rato - Introdução

Considere um labirinto como o do desenho abaixo, onde em alguma posição desse labirinto está um gato e em outra está uma toca onde há um rato. A intenção é ajudar o gato a encontrar o rato:



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. Definam tais valores como **VAZIO** e **PAREDE**, utilizando a diretiva **#define**.

Suponha que a cada passo o gato possa se deslocar uma casa na vertical ou uma na horizontal. Um método geral para resolver esse problema consiste em marcar com o número  $k$  ( $k = 1, 2, \dots$ ) as casas livres a partir da posição da toca do Rato. Para isso, rotula-se inicialmente a posição da toca do rato com  $k = 1$  e, para cada casa livre vizinha a esta, marca-se com  $k_{anterior} + 1 = 2$ . Por sua vez, marca-se com  $k_{anterior} + 1 = 3$  as casas vazias e vizinhas a estas últimas, e assim por diante. A marcação continua até que nenhuma casa esteja em condições de ser marcada.

Para o labirinto  $L = [1\dots 11, 1\dots 6]$  mostrado na figura, onde a toca do rato está na posição  $[11, 6]$ , será produzida a seguinte matriz ao final da marcação:

|    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|
| 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, caso exista, pode ser determinado, partindo da posição inicial do gato e passando, a cada etapa para uma casa vizinha, cuja numeração seja menor do que a atual.

**Exemplo:** se o gato estiver na posição  $[1, 2]$ , este precisará percorrer pelo menos 25 casas para chegar ao rato:

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

## 2 Programa

Faça um programa em C para resolver o seguinte problema:

São dados dois inteiros positivos  $m, n$ , uma matriz  $L = [1\dots m, 1\dots n]$ <sup>1</sup> que representa um labirinto, a posição  $(g1, g2)$  e  $(r1, r2)$  do gato e da toca do rato, respectivamente, nesse labirinto.

Caso exista, determinar o caminho mais curto que o gato deve percorrer até encontrar o rato, utilizando **obrigatoriamente** o método citado anteriormente, e imprimir em um arquivo de saída *.txt* o labirinto original, o labirinto marcado e o labirinto com desenho do caminho. Se tal caminho não existe, o programa deverá imprimir também em um arquivo *.txt* a matriz original, a matriz marcada, uma mensagem dizendo que será impossível o Gato encontrar o Rato e um desenho ilustrativo do labirinto mostrando a posição do Gato e do Rato<sup>2</sup>.

O seu programa **deve** fazer uso ao menos das seguintes funções, exatamente como elas estão aqui descritas:

```
/* Recebe a matriz labirinto L de inteiros, os ponteiros para os numeros inteiros de      *
 * colunas *m e linha *n de L, os ponteiros para inteiros da posicao *r1 e *r2 do rato *
 * e os ponteiros para a posicao *g1 e *g2 do gato. Esta funcao le o nome de um arquivo *
 * .txt, e adquire os valores de L, *m, *n, *r1, *r2, *g1 e *g2 deste arquivo          */
void le_labirinto (int L[][MAX_LIN], int *m, int *n, int *r1, int *r2, int *g1, int *g2);

/* Recebe o ponteiro *saida para o arquivo de saida, a matriz labirinto L de inteiros *
 * e os números (inteiros) de colunas m e linhas n. Esta funcao imprime L em *saida */
void imprime_labirinto_original (FILE *saida, int L[][MAX_LIN], int m, int n);

/* Recebe o ponteiro *saida para o arquivo de saida, a matriz labirinto L de inteiros, *
 * os números (inteiros) de colunas m e linhas n e a posicao r1 e r2 do rato. Esta      *
 * funcao marca em todos os possiveis espacos VAZIOS de L o numero de casas que falta *
 * para alcancar (r1, r2) e imprime L depois de marcado em *saida.                    */
void marca_labirinto (FILE *saida, int L[][MAX_LIN], int m, int n, int r1, int r2);

/* Recebe o ponteiro *saida para o arquivo de saida, a matriz labirinto L de inteiros, *
 * os números (inteiros) de colunas m e linhas n de L e os inteiros da posicao do      *
 * gato g1 e g2. Esta funcao imprime em *saida o caminho do gato e o labirinto com uma *
 * ilustracao desse caminho.                                                           */
void encontra_caminho (FILE *saida, int L[][MAX_LIN], int m, int n, int g1, int g2);
```

É possível resolver o problema apenas com as funções listadas. Implementar mais funções poderá apenas trazer complicação, mas vocês são livres para fazer o que considerarem melhor.

<sup>1</sup> $m$  representa o número de colunas e  $n$  o número de linhas.

<sup>2</sup>Mais informações sobre a impressão na seção 4.

### 3 Entrada

O seu programa deverá adquirir os dados a partir do arquivo de entrada *labirinto.txt*. O modelo padrão deste arquivo é:

```
11 6
11 6
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  $g2$  e depois o labirinto, onde **e** corresponde a um espaço vazio e **p** corresponde a uma parede (o caracter **\*** representa o fim do arquivo). Este é o padrão que deve ser seguido.

Existe um programa no paca que mostra como ler e escrever arquivos externos.

### 4 Saída

O seu programa deve criar um arquivo de saída com nome *caminho.txt*.

A saída do programa quando lê um labirinto, para o qual existe um caminho mais curto do Gato até o Rato deve ser:

```
>>> LABIRINTO ORIGINAL <<<
*****
*  ##      *
* #  #  ## *
* #  #  #*
* #  #  ## *
*  #  #  # *
* #      # *
*****

>>> LABIRITO MARCADO <<<
*****
* 26 27##### 12 11 10 9 10 11 12*
* 25###      ### 12### 8##### 13*
* 24 25###    ### 13### 7 6 5###*
* 23### 21### 15 14 15##### 4 3*
* 22 21 20### 16### 16 17 18### 2*
* 23### 19 18 17 18 17 18### 2 1*
*****
```

```
>>> LABIRITO COM CAMINHO <<<
```

```
*****
*  ##  xxx  *
*G#  #x#x## *
*x#  #x#xxx#*
*x#  #xx  ##xx*
*xxx#x#  #x*
*  #xxx  # x*
*****
```

Caso o caminho não exista a saída deverá ser:

```
>>> LABIRINTO ORIGINAL <<<
```

```
*****
*  ##          *
*  #  #  #  ## *
*  #  #  #  #*
*  #  #  ##  *
*  #  #  #  # *
*  #          # *
*****
```

```
>>> LABIRITO MARCADO <<<
```

```
*****
*      ##### 12 11 10  9 10 11 12*
*   ###      ### 12### 8##### 13*
*      ###   ### 13###  7  6  5###*
*   ### 21### 15 14 15#####  4  3*
*   ### 20### 16### 16 17 18###  2*
*   ### 19 18 17 18 17 18###  2  1*
*****
```

Nao eh possivel que o Gato encontre o Rato.

```
>>> LABIRITO COM CAMINHO <<<
```

```
*****
*  ##          *
*G#  #  #  ## *
*  #  #  #  #*
*  #  #  ##  *
*  #  #  #  # *
*  #          # R*
*****
```

## 5 Instruções

### 5.1 Sobre a Elaboração

O EP pode ser elaborado por equipes de dois alunos, desde que as seguintes regras sejam respeitadas:

- Os alunos devem trabalhar sempre juntos. A ideia é que deve existir cooperação.
- Caso em um grupo exista um aluno com maior facilidade, este deve explicar as decisões tomadas, e o seu parceiro deve participar e se esforçar para entender o desenvolvimento do programa. (denominamos isso de *programação em pares*, uma excelente prática que vocês devem se esforçar para adotar).
- Mesmo a digitação do EP deve ser feita em grupo, enquanto um digita, o outro acompanha.
- Recomendamos fortemente que o exercício seja desenvolvido da forma descrita nos itens acima, mas também pode ser realizado individualmente.

## 5.2 Sobre a Avaliação

- É de sua responsabilidade manter o código do seu EP em sigilo, ou seja, apenas você e seu parceiro podem ter acesso ao código.
- No caso de EPs realizados em dupla, a mesma nota será atribuída aos dois alunos do grupo.
- **Não serão toleradas cópias!** Exercícios copiados (com eventuais disfarces) levarão à reprovação do aluno na disciplina e o encaminhamento do caso para a Comissão de Graduação.
- Exercícios atrasados não serão aceitos.
- Exercícios com erro de sintaxe (ou seja, erros de compilação) receberão nota ZERO!
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. Isto irá influenciar a sua nota.
- Caso o programa apresente resultados "estranhos" (inesperados) para eventuais dados de entrada "incorretos", haverá desconto de nota.
- As informações impressas pelo seu programa deverão seguir **exatamente** o modelo pedido.
- Caso o seu programa não apresente alguma das especificações do enunciado ou das regras de envio, haverá desconto de nota.
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor avaliado ele será.

## 5.3 Sobre a entrega

- O prazo de entrega é até o dia 26/11/2010.
- Caso feito em dupla: **Ambos devem submeter a versão final do código.** Haverá desconto de nota para o membro da dupla que não submeter seu código.
- Entregar apenas um arquivo com o nome GatoRato.c. Haverá desconto de nota para extensões diferentes (.txt, .cpp, etc) e nomes diferentes do especificado.
- Para a entrega, utilize o Paca. Você pode entregar várias versões de um mesmo EP até o término do prazo, mas somente a última versão permanecerá armazenada pelo sistema.

- Não serão aceitas submissões por e-mail ou atrasadas. Não deixe para a última hora, pois o sistema pode ficar congestionado, e você corre o risco de não conseguir enviar.
- Guarde uma cópia do seu EP, pelo menos até o final do semestre.
- No início do arquivo, acrescente o seguinte cabeçalho. Haverá desconto de nota para ausência do cabeçalho, cabeçalho diferente ou incompleto:

```

/*****/
/** MAC 115 IFUSP Diurno 2010 **/
/** Prof Roberto Cesar **/
/** **/
/** Terceito Exercicio Programa -- Gato e Rato **/
/** **/
/** <nome do(a) aluno(a)> <numero USP> **/
/** <nome do(a) aluno(a)> <numero USP> **/
/** **/
/** Informacoes sobre Desenvolvimento: **/
/** <Ambiente (Dev-C++, CodeBlocks, Gedit, ...)> **/
/** <Sistema Operacional (Windows, Linux, ...)> **/
/*****/

```