



**Parte Prática do Exame da 2ª chamada de
Algoritmos e Programação**

22 de Fevereiro de 2003

Duração: 1h 30 m

Atenção:

Para cada um dos exercícios pedidos, faça o Mapa de Análise do Problema, o Algoritmo e implemente o respectivo Código em C.

As funções desenvolvidas devem ser portáteis.

A prova é sem consulta.

Faça uma pergunta em cada folha de prova e não escreva na 1ª página de cada uma das folhas.

1)

- a) Faça uma função que receba por argumento as coordenadas dos cantos opostos de um rectângulo e as coordenadas de um ponto. A função deve devolver um valor diferente de zero se o ponto pertencer ao rectângulo e deve devolver zero caso o ponto não pertença.

A função deve ter o protótipo seguinte:

```
int dentro(int x, int y, int c1x, int c1y, int c2x, int c2y);
```

onde os dois primeiros parâmetros são as coordenadas do ponto e os restantes parâmetros correspondem às coordenadas dos dois cantos opostos de um rectângulo.

- b) O pedaço de código abaixo possibilita determinar aleatoriamente as coordenadas dos cantos de um rectângulo:

```
void main(void)  
{  
    int c1x, c1y, c2x, c2y;  
  
    ...  
  
    //inicializar o gerador de números aleatórios  
    srand((unsigned)time(NULL));  
  
    c1x = rand() % LARGURA_MAX; //abscissa do canto 1  
    c1y = rand() % ALTURA_MAX; //ordenada do canto 1  
    c2x = rand() % LARGURA_MAX; //abscissa do canto 2  
    c2y = rand() % ALTURA_MAX; //ordenada do canto 2  
  
    ...  
}
```



Complete o programa anterior, de forma a construir um jogo que conta o número de tentativa que o utilizador demora até adivinhar as coordenadas de um ponto que pertença ao rectângulo.

O jogo termina quando o utilizador acertar, mostrando o número de tentativas gastas.

O programa tem de chamar (correctamente) a função definida na alínea anterior para determinar se o ponto está, ou não, dentro do rectângulo.

- 2) Faça uma função que transforme uma *string* (*array* de caracteres terminado por '\0') na *string* correspondente à concatenação da *string* inicial com a sua imagem num espelho. (ex.: *string* inicial “Ola Bom Noite ! ” *string* final “Ola Boa Noite ! ! etioN aoB alO”).

A função recebe como argumentos o *array* de caracteres, bem como o seu tamanho.

Nota: Caso necessite pode utilizar a função:

`int strlen(char str[]);`

a qual devolve o número de caracteres efectivos existentes na *string* “str” (sem contar com o '\0’).