Terceira Prova de Algoritmos e Estruturas de Dados I

Prof. Marcos Castilho

20 de março de 2003

Observações: A compreensão do enunciado faz parte da prova, que é individual e **sem** consulta e que, sendo um documento, deve ser feita à caneta. Qualquer fraude acarretará abertura de processo administrativo. Nos programas que você fizer será analisado: a lógica, o uso correto dos comandos, a sintaxe, a correta declaração dos tipos e os nomes das variáveis, a endentação, a clareza e a criatividade, além, é claro, do correto uso de procedimentos e funções, incluindo a correta passagem de parâmetros.

1. (10 pontos) Considere em PASCAL a declaração: `TYPE palavra=string[20];`
   Implemente uma função que receba como parâmetros duas variáveis do tipo palavra e retorne -1 se a primeira palavra for lexicograficamente menor que a segunda, 0 se forem iguais, e +1 se a primeira for lexicograficamente maior que a segunda.

2. (20 pontos) Considere o problema do "Jogo da Vida" estudado em sala de aula. Sem qualquer sombra de dúvidas, a operação mais cara computacionalmente falando em qualquer algoritmo que implemente a solução do jogo é a contagem do número de vizinhos de cada célula. Considerando uma célula fora da borda do tabuleiro, escreva um trecho de código PASCAL que obtenha o número de vizinhos de uma célula. Você **não** pode fazer uso de **nenhum** comando *for*, *repeat*, *while*, *if*, ou *case*. Este trecho de código deverá conter apenas e tão somente a definição do tipo de dados para o tabuleiro e os comandos para contar o número de vizinhos.

3. (10 pontos) No item anterior foi solicitado para não se considerar a borda do tabuleiro. Nesta questão você deve indicar, segundo a estrutura de dados definida no item anterior, qual o comportamento do seu programa quando se tenta contar o número de vizinhos de uma célula da borda. Se por acaso houver alguma possibilidade de erros, indique como deve ser modificada a estrura de dados para que o erro deixe de existir.

4. (30 pontos) Aqui temos uma forma peculiar de realizar uma multiplicação entre dois números: divida o primeiro por 2 e multiplique o segundo por 2 até que o primeiro seja reduzido a 1. Toda vez que o primeiro for impar, lembre-se do segundo. Não considere qualquer fração durante o processo. O produto dos dois números é igual a soma dos números que foram lembrados. Exemplo: $53 \times 26 =$

53	26	13	6	3	1
26	52	104	208	416	832
26 +		104 +		416 +	832 = 1378

   Faça uma função em PASCAL que receba dois inteiros e retorne o produto dos dois, do modo como foi especificado acima. Não é permitido uso de array.

5. (30 pontos) Considere o tipo abstrato de dados lista definido em aula:

```
TYPE
    TipoIndice = 1..TAM_MAX_LISTA;
    TipoElemento = RECORD
                        Num : Real;
                        Prox: TipoIndice;
                   END;
    Lista = RECORD
                 Tam: integer;
                 Primeiro: TipoIndice;
                 Vet: ARRAY [1..TAM_MAX_LISTA] of TipoElemento;
            END;
```

Considerando que os elementos da lista são inseridos em ordem crescente, implemente uma função InsereOrdenado em PASCAL que receba como parâmetros uma lista do tipo lista e um elemento real e insira este elemento no local correto na lista ordenada.