Universidade Estadual de Campinas
Algoritmos e Programação de Computadores (MC102 Turma Z)

Lista de Exercícios 8

1. Implemente as funções especificadas abaixo de forma recursiva:
   a) int SomaInteiros(int n);
   Calcula a soma de todos os números inteiros positivos menores que n.

   b) void PulaLinhas(int n);
   Imprime n linhas em branco na saída padrão.

   c) void ImprimePiramide(int n);
   Imprime uma pirâmide conforme o exemplo abaixo de forma recursiva onde o maior valor n é fornecido como argumento.
   Ex: Para n=5 temos:
   1
   2 1
   3 2 1
   4 3 2 1
   5 4 3 2 1

   d) void DigitosInvertidos(int n);
   Imprime os dígitos de um número inteiro n em ordem inversa.
   Ex: Para n=123 deve imprimir 321.

   e) int ContaDigitos(int n);
   Retorna o número de dígitos do valor n fornecido.
   Ex: Para n=450 a função deve retornar 3.

   f) float Potencia(float x, int n);
   Calcula x^n para qualquer x real e n inteiro, inclusive para valores de n negativos.

   g) int Fibonacci(int n);
   Calcula de forma recursiva o n-éssimo número da seqüência de Fibonacci.
   A seqüência de Fibonacci consiste em uma série de números, tais que, definindo seus dois primeiros números como sendo 0 e 1, os números seguintes são obtidos através a soma dos seus dois antecessores.
   Exemplos da seqüência são 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...

   h) float CalculaMedia(float V[], int n);
   Calcula de forma recursiva a média dos elementos de um vetor de tamanho n.

   i)Escreva uma função recursiva que analisa os elementos de um vetor e retorna um dentre os seguintes códigos:
   Código Condição do vetor
   0 Elementos desordenados
   1 Elementos em ordem crescente
   2 Elementos constantes
   3 Elementos em ordem decrescente
   int AnalisaVetor(int V[], int n);
   Use o protótipo acima onde “n” indica o número de elementos presentes no vetor.

Exemplos:
Para V={1,2,2,5,6,6,7,8,8,9} retorna código 1.
Para V={20,15,11,10,8,8,5,2} retorna código 3.
Para V={1,20,2,5,6,6,7,80,9} retorna código 0.
Para V={8,8,8,8,8,8,8,8,8,8} retorna código 2.

2. Faça uma função recursiva MaxMin que calcula o elemento máximo e o elemento mínimo de um vetor com *n* números inteiros.

3. Faça uma função recursiva Dígito que recebe um número inteiro *n* e calcula a soma dos digitos de *n*. Exemplo: se *n* = 32 então Dígito(*n*) = 6.

4. Faça uma função recursiva que verifica se uma expressão de parênteses é bem formada (uma expressão de parênteses é dita bem formada se, para cada parêntese de abertura, existe um parêntese de fechamento após ele na expressão). Assuma que apenas os parênteses da expressão serão digitados, e nenhum outro tipo de caracter. O que muda no algoritmo se a expressão possuir dígitos e operadores (+, -, * e /) também?