UFMS-CCET-DCT-Algoritmos e Estruturas de Dados 1 2004
Lista 6 - Entregar até 25/08 às 19h na Secretaria do DCT

Prof. Marco Aurélio

6 de agosto de 2004

1. Considere os seguintes módulos:

```
// Troca o conteudo de x e de y entre si.
Função inteiro TROCA ( REF inteiro x, inteiro y)
        aux ← x
        x ← y
        y ← aux
FimFunção
// Devolve o maior dos valores entre a, b e c.
Função inteiro MAIOR ( REF inteiro a, inteiro b, inteiro c)
        se b > a então
           a ← b
        fimse
        se c > a então
           a ← c
        fimse
retorne a
FimFunção
```

   Cada um dos módulos é precedido por um comentário do que o módulo deveria fazer. Indique os erros na definição dos módulos.

2. (a) Escreva uma função de nome MENORES com três parâmetros de entrada:
   - um valor inteiro x;
   - um vetor de inteiros V;
   - um inteiro n ($1 \leq n \leq 100$), indicando o tamanho de V.

   O valor de retorno da função MENORES é a quantidade de inteiros em V menores ou iguais a x.

   (b) Escreva um algoritmo que, dado um conjunto de n números inteiros, mostre estes números em ordem não decrescente de valor. A entrada do algoritmo consiste no valor de n ($1 \leq n \leq 100$) e dos n números inteiros. Seu algoritmo deve usar a função MENORES.

3. Considere um tipo chamado vetcadeia definido como:

```
DefinaTipo vetor[0..99] de caracter vetcadeia
```

   O tipo vetcadeia pode armazenar uma cadeia de até 100 caracteres. Para controlar a quantidade de caracteres efetivamente armazenadas, vamos utilizar o código 0 como um caracter específico para esta finalidade, o qual é colocado no vetor na posição imediatamente após a cadeia de caracteres armazenada no vetor. Uma vez que o tipo vetcadeia é um vetor com índice inicial 0, a posição de 0 indica o tamanho da cadeia armazenada.

   (a) Construa uma função de nome Tamanho com um parâmetro de entrada s, que é uma variável do tipo vetcadeia, e devolve o tamanho da cadeia armazenada em s.

(b) Construa uma função de nome MaiorPrefixo com dois parâmetros de entrada do tipo vetcadeia, $s1$ e $s2$, e devolve o tamanho do maior prefixo comum de $s1$ e $s2$.

(c) Construa uma função de nome Compara com dois parâmetros de entrada do tipo vetcadeia, $s1$ e $s2$, e devolve um dos seguintes valores:

- 0, caso $s1 = s2$;
- um valor maior que 0, caso o tamanho da cadeia em $s1$ seja maior que o de $s2$;
- um valor menor que 0, caso o tamanho da cadeia em $s1$ seja menor que o de $s2$.

(d) Construa um procedimento Copia com um parâmetro de entrada $s1$ e um parâmetro de saída $s2$, ambos do tipo vetcadeia, para copiar a cadeia em $s1$ para $s2$.

4. Um número a é dito permutação de um número b se os dígitos de a formam uma permutação dos dígitos de b.
Exemplo: 5412434 é uma permutação de 4321445, mas não uma permutação de 4312455.
Obs.: Considere que o dígito 0 (zero) não aparece nos números.

(a) Faça uma função conta_digitos que dados um inteiro n e um inteiro $d, 0 < d < 9$, devolve quantas vezes o dígito d aparece em n.

(b) Usando a função do item anterior, faça um algoritmo que lê dois números a e b e responda se a é permutação de b.

5. (a) Faça uma função arctan que recebe o número real $x \in [0; 1]$ e devolve uma aproximação do arco tangente de x (em radianos) através da série incluindo todos os termos da série

$$arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$

incluindo todos os termos da sérei até $|\frac{x^k}{k}| < 0,0001$

(b) Faça uma função angulo que recebe um ponto de coordenadas cartesianas reais (x, y), com $x > 0$ e $y > 0$ e devolve o ângulo formado pelo vetor (x, y) e o eixo horizontal.

(c) Use a função do item anterior mesmo que você não a tenha feito. Note que a função só calcula o arco tangente de números entre 0 e 1, e o valor devolvido e o ângulo em radianos (use o valor = 3.14 radianos = 180 graus). Para calcular o valor do ângulo pedido, use a seguinte fórmula:

$$\alpha = \begin{cases} \arctan(y/x), & \text{caso } y < x, \\ \frac{\pi}{2} - \arctan(x/y), & \text{caso contrário} \end{cases} \quad (1)$$

(d) faça um algoritmo que, dados n pontos do primeiro quadrante ($x > 0$ e $y > 0$) através de suas coordenadas cartesianas, determina o ponto que forma o menor ângulo com o eixo horizontal. Use a função do item anterior, mesmo que você não a tenha feito.

6. (a) Escreva uma função que calcula a soma dos elementos da linha i de uma matriz real $A_{m \times n}$.

(b) Escreva uma função que calcula o produto dos elementos da coluna j de uma matriz real $A_{m \times n}$.

7. (a) faça uma função MAX que recebe como entrada um inteiro n, uma matriz inteira $A_{m \times n}$ e devolve três inteiros: k, Lin e Col. O inteiro k é um maior elemento de A e é igual a $A[Lin, Col]$.

(b) faça um algoritmo que, dado um inteiro n e uma matriz quadrada de ordem n, cujos elementos são todos inteiros positivos, imprime uma tabela onde os elementos são listados em ordem decrescente, acompanhados da indicação de linha e coluna a que pertencem. Havendo repetições de elementos na matriz, a ordem é irrelevante. Utilize obrigatoriamente o procedimento da parte anterior, mesmo que você não o tenha feito.

8. Para encontrar uma raiz de um polinômio $p(x) = a_0 + a_1 x + \ldots + a_n x^n$, $(n > 2)$, pode-se aplicar o método de Newton, que consiste em re nar uma aproximação inicial x_0 dessa raiz através da expressão:

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}$$

$n = 0, 1, 2, \ldots$, onde $p'(x)$ é a primeira derivada de $p(x)$.

Usualmente, repete-se esse refinamento até que $|x_{n+1} - x_n| < \epsilon$, $\epsilon > 0$, ou até que m iterações tenham sido executadas. Dados um polinômio $p(x) = a_0 + a_1 x + \ldots + a_n x^n$, uma aproximação inicial x_0 da raiz de $p(x)$, $\epsilon > 0$ e o número máximo de iterações que devem ser executadas, determine uma aproximação da raiz de $p(x)$ pelo método de Newton. Utilize uma função que, dado um polinômio $p(x)$, calcula a derivada $p'(x)$ e, para calcular $p(x_n)$ e $p'(x_n)$ em cada iteração, uma função que calcula o valor de um polinômio em um ponto.