UFMS-CCET-DCT-Algoritmos e Estruturas de Dados 1 2004
Lista 6 - Entregar até 29/08 às 19h na Secretaria do DCT

Prof. Marco Aurélio

24 de agosto de 2005

1. (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.

2. Considere um tipo chamado vetcadeia definido como:

   <u>DefinaTipo</u> vetor[0..99] de <u>caracter</u> 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 $s2$.

3. (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:

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

(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.

4. (a) faça uma função MAX que recebe como entrada um inteiro n, uma matriz inteira $A_{n\times n}$ 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.

5. Para encontrar uma raiz de um polinômio $p(x) = a_0 + a_1x + \ldots + a_nx^n$, $(n > 2)$, 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 $p(x) = a_0+a_1x+\ldots+a_nx^n$, 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.

6. O que fará cada um dos algoritmos abaixo quando executados?

```
função de inteiro A(literal ^s, ^t)

    inteiro i

    i ← 1
    enquanto ^s=^t faça
        s ← s + 1
        t ← t + 1
        i ← i + 1
    fimenquanto
    devolva i
fimfunção
```

```
Algoritmo B

    vetor[0:5] de inteiro A
    inteiro i, ^p

    p ← A
    para i de 0 até 5 faça
        leia A[i]
    fimpara
    para i de 0 até 5 faça
        escreva ^p+i
    fimpara
fimalgoritmo
```

```
Algoritmo C

    inteiro a, b, ^p, ^^pp

    a ← 10
    b ← 13
    p ← &b
    pp ← &p
    ^^pp ← ^p+b
    ^pp ← &a
    ^^pp← ^p+a
    escreva a, b, ^p, ^^pp
fimalgoritmo
```

7. escreva uma função que receba um vetor de inteiros A de tamanho indefinido contendo somente valores positivos e na última posição o valor 0 indicando o fim do vetor, calcule para todo i o número de elementos de A iguais a $A[i]$ e devolva o vetor A com estes valores calculados e armazenados em $A[i]$. Note que você deve usar alocação dinâmica de um vetor auxiliar.

8. No *Bublle-sort*, ao final de cada passo i, x_{n-i+1} se encontra em sua posição correta. Reescreva o algoritmo de forma a tirar profeito deste fato. Qual a complexidade de pior caso desta nova versão?

9. Execute passo a passo o *Insertion-sort* para a seguinte entrada:

$$23, 38, 12, 19, 06, 54, 26, 03$$

10. Execute passo a passo o *Selection-sort* para a entrada acima.