
MC102—Algoritmos e Programação de Computadores

Lista de exercícios 02

Turmas L, O, T e U — 2º semestre de 2005

1. Escreva um programa que lê uma lista de inteiros positivos e imprime essa lista em ordem inversa. A lista pode ter até 100 elementos e a leitura deve ser interrompida assim que o primeiro inteiro não positivo for digitado. Não será informado previamente o número de inteiros na lista de inteiros.
2. Escreva um programa que leia as coordenadas de um vetor de dimensão n , n informado pelo usuário, e calcule o módulo desse vetor. O módulo de um vetor de dimensão n é calculado da seguinte forma:

$$\text{modulo} = \sqrt{\sum_{k=1}^n \text{coordenada}_k^2}$$

3. Escreva uma função que, dado um vetor de n posições, calcule o número de elementos pares nesse vetor. O cabeçalho da sua função deve ser:

```
int pares (int vetor [], int n);
```
4. Escreva uma função que, dado uma matriz quadrada de dimensão 5×5 , imprima a sua transposta. O cabeçalho da função deve ser

```
int mostra_transposta (int matriz[5][5]);
```
5. Escreva um procedimento que, dado uma matriz $n \times n$, preencha essa matriz de forma que ela torne-se a matriz identidade de tamanho n (I_n). O tamanho máximo da matriz é limitado pela constante LIM e o cabeçalho da função deve ser:

```
void identidade (int matriz[LIM][LIM], int n);
```
6. Escreva uma função que calcule o fatorial de um número. Seu cabeçalho deve ser:

```
int fatorial (int n);
```
7. Escreva uma função que calcula o número de combinações entre n elementos distintos agrupados em conjuntos de p elementos, onde

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}$$

Lembre-se que p deve estar entre 0 e n . Você pode utilizar a função fatorial do exercício anterior. O cabeçalho da função deve ser:

```
int binomial (int n, int p);
```

8. Escreva um procedimento que intercala os valores de dois vetores inteiros de mesmo tamanho em um terceiro vetor. Assuma que, por construção, a dimensão do terceiro vetor é pelo menos duas vezes a dimensão de cada um dos outros vetores. Seu cabeçalho deve ser:

```
void intercala (int v1[], int v2[], int v3[], int dimensao);
```

9. Escreva uma função que, dado um vetor de dimensão n , retorne o índice do maior elemento daquele vetor. O cabeçalho da função deve ser:

```
int indice_maximo (int v[], int n);
```

10. Escreva um programa que, utilizando a função do exercício anterior, faça a leitura de um vetor e mostre o maior elemento desse vetor.

11. Altere o programa do exercício anterior para que ele mostre todos os elementos do vetor cujo valor é máximo (ou seja, se existirem dois ou mais elementos desse vetor com o valor máximo, todos deverão ser exibidos).

12. Repita os três últimos exercícios, agora para o valor mínimo.

13. Baseado na função que encontra o valor máximo de um vetor, como você faria para ordenar esse vetor? Faça um programa em linguagem C que faça essa ordenação.

14. Implemente um programa que compare o módulo de dois vetores de dimensões iguais e retorne -1 se o módulo do primeiro vetor for menor, 1 se o módulo do segundo vetor for maior e 0 caso os módulos sejam iguais. Você pode utilizar a função módulo, implementada anteriormente.

15. Dado o tipo `diahora`, como descrito abaixo

```
struct diahora {  
    int dias;  
    int horas;  
    int minutos;  
    int segundo;  
}
```

Escreva uma função que receba como parâmetro o número de segundos e retorne o número de dias, horas, minutos e segundos que corresponde a aquela quantidade de segundos. Sua função deve ter o seguinte cabeçalho:

```
struct diahora dias (int segundos);
```

16. Dado o vetor ordenado $v = \{1, 5, 6, 7, 12, 17, 22, 45, 65, 98, 100, 120, 150, 200, 231\}$. Mostre passo a passo a pesquisa binária pelo elemento 100. Quantas comparações foram feitas? E se a pesquisa fosse sequencial, quantas comparações seriam feitas?
17. Para o vetor acima, dê um exemplo onde a pesquisa sequencial faria menos comparações que a pesquisa binária.
18. Ordene o vetor $v = \{20, 12, 28, 05, 10, 18\}$ usando os métodos de inserção, seleção e bolha. Mostre o vetor a cada passo do *loop*, para cada um dos métodos.
19. Transforme o algoritmo de pesquisa sequencial visto em sala de aula em uma função de pesquisa sequencial. Repita o mesmo para todas as funções de pesquisa e ordenação.
20. **Calculando raiz quadrada:** Os computadores (e qualquer espécie de equipamento eletrônico) somente conseguem realizar as quatro operações aritméticas básicas. Combinando essas operações é possível calcular funções mais complexas, como seno, cosseno e a raiz quadrada. Para calcular a raiz quadrada, os computadores calculam iterativamente, até conseguir um valor próximo o suficiente da raiz quadrada, mas nunca o valor exato.

O cálculo é interrompido quando o erro é pequeno o suficiente para não ser percebido pelo usuário (ou seja, o erro não pode ser mais representado dentro de uma variável ponto flutuante). Ao atingir esse estado, apesar de ser possível calcular um valor mais preciso para a raiz quadrada, os computadores não podem mais armazená-lo.

Para o cálculo da raiz quadrada de um determinado valor n , a equação iterativa é:

$$x_{i+1} = \frac{1}{2} \left(x_i + \frac{n}{x_i} \right)$$

onde x_i é o valor atual da raiz quadrada e x_{i+1} é o próximo valor da raiz quadrada. A primeira raiz (x_0) pode ser um valor qualquer, preferencialmente próximo da raiz quadrada (metade do valor já é uma boa aproximação). Escreva uma função em C que, dado um número ponto flutuante, retorne a sua raiz quadrada.

21. Escreva uma função que, dado um polinômio na forma vetorial, onde o i -ésimo elemento corresponde ao coeficiente de x^i , calcule a sua derivada. Tanto o algoritmo quanto a representação de polinômios em vetores já foi vista em sala de aula. Sua função não pode alterar os valores do vetor de entrada (Dica: utilize dois vetores).
22. Escreva uma função que, dado um polinômio na forma vetorial, calcule todas as derivadas até que a última derivada tenha valor igual a zero. Novamente, o vetor original não pode ser alterado (Dica: utilize uma matriz para retornar todas as derivadas).
23. Escreva uma função que, dado um polinômio na forma vetorial e um número real x , calcule $f(x)$ (o valor do polinômio para aquele valor de x).

24. **Zero de funções:** O método de cálculo da raiz quadrada é derivado de um método mais geral, para cálculo de zero de funções. Novamente, inicia-se o cálculo com x_0 qualquer, mas preferencialmente próximo do zero que se deseja obter, e calcula-se os próximos valores da seguinte forma

$$x_{i+1} = \left(x_i - \frac{f(x_i)}{f'(x_i)} \right)$$

Escreva um programa que, dado um polinômio (na forma de um vetor, como visto em sala de aula) e um ponto inicial, calcule um possível zero desse polinômio.

25. **Série de Taylor:** O valor de toda função em um determinado ponto pode ser representado como um somatório, onde x é o valor que se deseja encontrar e a é um ponto conhecido.

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n$$

Escreva duas funções em linguagem C, $\sin(x)$ e $\cos(x)$, que calculam respectivamente, o seno e o cosseno de um valor x , tal que $x \in [-2\pi, 2\pi]$. Não se esqueça que $\sin(0) = 0$ e $\cos(0) = 1$ e Como você faria para calcular o seno e o cosseno para qualquer valor de x ?

26. **Integral numérica:** É muito complicado obter uma integral exata para determinados tipos de funções. Quando desejamos somente o valor da integral, e não sua forma analítica, pode ser útil calcular a integral por aproximação. A fórmula de Simpson permite-nos aproximar o valor de uma determinada integral se soubermos calcular os pontos da função original utilizando a expressão abaixo:

$$\int_a^b f(x)dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + 4f(x_{n-1}) + f(x_n)]$$

Onde $h = \frac{(b-a)}{n}$, $x_i = a + ih$ e n deve ser um número par. Suponha que exista em seu programa uma função `float f(float x)`, que retorna o valor de f para cada valor de x . Escreva um programa que calcule a integral de $f(x)$ utilizando a fórmula de Simpson considerando $n = 10$.

27. Reescreva o seu programa para que ele calcule a integral utilizando um número de passos variáveis (ou seja, n agora deve ser informado pelo usuário). Lembre-se de testar se n é um número par.