

MC102—Algoritmos e Programação de Computadores

Turmas A e B

Primeiro semestre de 2010

Lista de exercícios 02 - Vetores e matrizes

1. Reescreva o programa a seguir realizando, a cada leitura, um teste para ver se a dimensão do vetor não foi ultrapassada. Caso o usuário entre com 100 números, o programa deverá abortar o *loop* de leitura automaticamente. O uso do Flag não deve ser retirado.

```
#include <stdio.h>
int main ()
{
    int num[100]; /* Declara um vetor de inteiros de 100 posicoes */
    int count=0;
    int totalnums;
    do
    {
        printf ("\nEntre com um numero (-999 p/ terminar): ");
        scanf ("%d",&num[count]);
        count++;
    } while (num[count-1]!=-999);
    totalnums=count-1;
    printf ("\n\n\n\t Os números que você digitou foram:\n\n");
    for (count=0;count<totalnums;count++)
        printf (" %d",num[count]);
    return 0;
}
```

2. Escreva um programa que leia um vetor de tamanho n da entrada padrão (n informado pelo usuário) e escreva na saída padrão o índice do maior elemento e o índice do menor elemento desse vetor.
3. 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:

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

4. 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.
5. Escreva um programa que leia da entrada um vetor de n posições (n fornecido pelo usuário) e mostre na saída o número de elementos pares nesse vetor.
6. O que faz o programa a seguir? Utilize um teste de mesa para seu auxílio.

```

#include <stdio.h>

int main()
{
    int vetor[100], n, i, p, j, aux;

    printf("Entre com um valor para n [1,100]: ");
    scanf("%d", &n);
    for(i = 0; i < n; i++){
        printf("Vetor[%d]: ", i);
        scanf("%d", &vetor[i]);
    }

    for(i = 0; i < n; i++){
        p = i;
        for(j = i + 1; j < n; j++){
            if(vetor[p] < vetor[j])
                p = j;
        }
        aux = vetor[p];
        vetor[p] = vetor[i];
        vetor[i] = aux;
    }

    printf("Resultados: %d", vetor[0]);
    for(i = 1; i < n; i++)
        printf(", %d", vetor[i]);

    return 0;
}

```

7. Escreva um programa que calcula a soma de dois polinômios de grau n . O seu programa deve ler primeiramente o grau n de cada polinômio e em seguida os $n + 1$ coeficientes de cada um dos dois polinômios, ordenados do coeficiente de maior grau para o coeficiente de menor grau em cada polinômio. O programa deve escrever na tela o grau do polinômio resultante e os coeficientes, ordenados do maior grau para o menor grau. Considere que o grau máximo do polinômio é 25 e que somente valores válidos são digitados. Um exemplo de execução seria:.

```

$./pol
5 2
5.0 2.0 3.3 2.0 4.5 5.2
3.0 -2.5 2.8
Grau do polinomio: 5
Polinomio Resultante: 5.0 2.0 3.3 5.0 2.0 8.0

```

8. Escreva um algoritmo e depois um programa em linguagem C que leia um conjunto de pares de números, um inteiro e um real, que representam número de matrícula e a nota do aluno. Calcule e imprima:
- a média da turma;
 - quantos alunos obtiveram nota maior do que a média;
 - quantos alunos obtiveram nota menor do que a média;
 - quantos alunos obtiveram nota igual à média;
 - qual foi a maior nota e quantos e quais alunos a obtiveram.

9. Faça um programa que inverta uma string, tamanho máximo 30: leia a string e armazene-a invertida em outra string. Use o comando for para varrer a string até o seu final.
10. Palíndromo é a frase ou palavra que mantém o mesmo sentido quando lida de trás pra frente. Faça um programa que informe se a palavra passada digitada é um palíndromo. Veja alguns exemplos abaixo. Note que são válidos apenas os caracteres, sinais de pontuação não são considerados na verificação de um palíndromo. Assuma que todos os caracteres serão minúsculos. Leia uma string sem pontuações e espaços e brancos. A string “a base do teto desaba?” ficaria “abasedotetodesaba”. (Observação: tente fazer sem retirar os espaços e pontuação, mas como especificado já é o bastante.)
11. A gerente do salão de cabeleireiros *Sempre Bela* tem uma tabela que registra os serviços de pinturas de pés, de mãos e o serviço de podologia realizados por cada uma de suas cinco manicures. Sabendo que cada uma ganha 50% do que faturou no mês, crie um programa em C que possa calcular e imprimir quanto cada uma vai receber no final do mês. Os preços cobrados pelos serviços são respectivamente R\$10,00 , R\$15,00 e R\$30,00.
12. Escreva um programa que leia um valor n da entrada padrão e preencha uma matriz de forma que ela torne-se a matriz identidade de tamanho n (I_n) e depois imprima essa mesma matriz na saída padrão. Considere que o usuário nunca digitará valores de n maiores que 100.
13. Escrever um programa que determine se uma matriz quadrada de $n > 0$ é uma matriz permutação. Uma matriz quadrada é chamada de matriz permutação se seus elementos são apenas 0's e 1's e se em cada linha e coluna da matriz existe apenas um único valor 1. Exemplo: A matriz abaixo é uma matriz permutação.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

14. Escreva um programa que leia da entrada padrão uma matriz de tamanho $n \times m$ (n e m fornecidos pelo usuário) e escreva na saída padrão o valor máximo e o valor mínimo existentes naquela matriz. Considere que o usuário nunca fornecerá n e m maiores que 100.
15. Escreva um programa que leia uma matriz quadrada de dimensões $n \times n$ (n fornecido pelo usuário), imprima a sua transposta. Considere que o usuário nunca digitará valores de n maiores que 100.
16. Um método conhecido de resolução de sistema de equações lineares é o chamado método de eliminação de Gauss. A idéia base desse método é transformar o sistema original $Ax = b$ num outro sistema equivalente $\tilde{A}x = \tilde{b}$, onde \tilde{A} é uma matriz triangular superior. O sistema equivalente será muito fácil de resolver pois bastará usar substituição ascendente (de “baixo para cima” calcular o valor de cada incógnita) para obter a solução pretendida.

Assim, o objetivo é obter uma matriz triangular superior a partir da matriz ampliada do sistema A , efetuando apenas operações elementares sobre as linhas da matriz ampliada.

Suponha que o processo para encontrar a matriz triangular superior foi resolvida por um aluno de MC202. Você deverá então construir um programa que leia tal matriz triangular de tamanho n e depois encontrar o valor das incógnitas através de substituição ascendente. Exemplo:

$$\begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} = \begin{pmatrix} 12 \\ 34 \\ 27 \\ -38 \end{pmatrix} \leftrightarrow \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} = \begin{pmatrix} 12 \\ 10 \\ -9 \\ -3 \end{pmatrix}$$

Note que resolver a linha i da matriz implica em encontrar o valor da incógnita x_i . Assim, $x_4 = -3 / -3 = 1$ e, ao substituir o valor de x_4 na equação da linha 3 da matriz, obteremos que $2x_3 - 5(x_4) = -9 \leftrightarrow 2x_3 - 5(1) = -9$, ou seja, $2x_3 = -4$ e $x_3 = -2$. Ao substituir nas outras equações, note que $x_2 = -3$ e $x_1 = 1$.