

1. Crie uma função que receba como parâmetro uma matriz estática de 5 linhas e 5 colunas, e a alimente com valores lidos do teclado. Faça ainda uma outra função que leia esta mesma matriz e exiba os seus elementos em tela, pulando as linhas. Finalize o seu programa criando uma função que retorne o maior elemento da matriz. Obedeça os seguintes protótipos.

```
void alimentar( int matriz[5][5] );  
void exibir( int matriz[5][5] );  
int maior( int matriz[5][5] );
```

2 - Modifique o programa acima, acrescentando uma nova função que retorne o maior elemento de cada linha armazenado em um vetor passado por parâmetro. Ou seja, o elemento vetor[0] irá armazenar o maior elemento da linha 0 da matriz passada como parâmetro. Obedeça o protótipo:

```
void maior_por_linha( int matriz[5][5], int vetor[5] );
```

Dada uma matriz real *matriz* com *m* linhas e *n* colunas e um vetor real *vetor* com *n* elementos, determinar o produto de *matriz* por *vetor*. Obedeça o protótipo:

```
void prod_mat_vet(int matriz[5][5], int vetor[5], int m, int n);
```

3 - Dadas duas matrizes reais  $A_{m \times n}$  e  $B_{n \times p}$ , calcular o produto de *A* por *B*, armazenando o resultado em *C*. Veja o protótipo:

```
void prod_mat(int **A, int **B, int **C, int m, int n, int p);
```

4 - Dizemos que uma matriz inteira  $A_{n \times n}$  é uma *matriz de permutação* se em cada linha e em cada coluna houver *n-1* elementos nulos e um único elemento igual a 1.

Exemplo: A matriz abaixo é de permutação:

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

Dada uma matriz inteira  $A_{n \times n}$ , verificar se *A* é de permutação. Crie o seu protótipo usando apenas matrizes estáticas.

5 - Dizemos que uma matriz quadrada inteira é um *quadrado mágico* se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais.

6 - Imprimir as *N* primeiras linhas do triângulo de pascal.

**7** – Faça um programa que lê um inteiro  $n < 100$  e os elementos de uma matriz real quadrada  $An \times n$  e verifica se a matriz A tem uma linha, coluna ou diagonal composta apenas por zeros.

**8** – Faça um programa que lê os elementos de uma matriz quadrada  $An \times n$  e retorna um vetor com a soma de cada linha. O elemento  $i$  do vetor V corresponde à soma de todos os números da matriz A na linha  $i$ . Use o protótipo abaixo.

```
void somaLinha(int matriz[MAX][MAX], int linhas, int colunas, int vetor[]);
```

**9** – Faça uma função que recebe uma matriz como parâmetro e retorna a soma de todos os elementos da diagonal secundária.

**10** – Faça uma função que recebe uma matriz como parâmetro e verifica se essa matriz possui duas linhas ou duas colunas onde cada elemento  $i$  de uma linha ou coluna é múltiplo do mesmo elemento  $i$  da outra linha ou coluna.

**11** – Faça uma função que recebe uma matriz e um inteiro como parâmetro e retorna uma matriz que é multiplicação da matriz recebida como parâmetro pelo inteiro.

**12.** Dado  $0 < n < 200$  e uma matriz real  $An \times n$ , verificar se A é simétrica.

Uma matriz  $An \times n$  é simétrica se, e somente se, A é igual a sua transposta. Neste caso, temos que verificar se cada  $A[\text{lin}][\text{col}]$  é igual a  $A[\text{col}][\text{lin}]$  como indicado na figura.