



Exercícios – Funções e Matrizes

Funções Utilitárias:

Implemente funções utilitárias que devem ser empregadas como parte das soluções dos demais exercícios nesta lista. A primeira delas deve ler do usuário os elementos de uma matriz. Para uma matriz de inteiros, uma possível assinatura poderia ser:

```
void le_mat(int mat[][MAX], int n_linhas, int n_colunas);
```

A segunda função utilitária deve escrever todos os elementos de uma matriz na tela. Para uma matriz de inteiros, uma possível assinatura poderia ser:

```
void imprime_mat(int mat[][MAX], int n_linhas, int n_colunas);
```

Questões:

1. Implemente uma função que receba uma matriz de números inteiros quadrada de ordem n como parâmetro de saída. A função a ser implementada deve armazenar na matriz uma matriz identidade de tamanho $n \times n$. A função `main` deve ler a ordem n da matriz quadrada e exibir a matriz resultante na tela utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas/colunas:
```

```
4
```

```
Matriz resultante:
```

```
1 0 0 0
```

```
0 1 0 0
```

```
0 0 1 0
```

```
0 0 0 1
```

2. Implemente uma função que receba uma matriz de números inteiros quadrada de ordem n como parâmetro de saída. A função a ser implementada deve armazenar em sua diagonal secundária um elemento x passado como parâmetro. A função `main` deve ler a ordem n da matriz quadrada, o número inteiro x e exibir a matriz resultante na tela utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas/colunas:
```

```
3
```

```
Informe o nr. a ser inserido na diagonal secundaria:
```

```
8
```

```
Matriz resultante:
```

```
0 0 8
```

```
0 8 0
```

```
8 0 0
```

3. Implemente uma função que receba uma matriz de números inteiros de tamanho $n_l \times n_c$ como parâmetro de saída. A função a ser implementada deve armazenar os $n_l \times n_c$ primeiros números ímpares nos seus elementos, seguindo a ordem de preencher cada coluna primeiro para então prosseguir para a próxima linha. A função `main` deve ler as dimensões `n_l` e `n_c` da matriz e exibir a matriz resultante na tela utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas da matriz:
2
Informe o nr. de colunas da matriz:
3
Matriz resultante:
1 5 9
3 7 11
```

4. Implemente uma função que receba uma matriz de números inteiros quadrada de ordem `n` como parâmetro de entrada e saída. A função a ser implementada deve transformar a matriz passada como parâmetro em uma matriz triangular superior: matriz em que os elementos M_{ij} abaixo da diagonal principal são iguais a 0. A função `main` deve ler a ordem `n` da matriz quadrada, cada um dos elementos da matriz e exibir a matriz resultante na tela utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas/colunas:
3
Informe os elementos da matriz:
1 -1 2
2 45 3
7 11 14
Matriz resultante:
1 -1 2
0 45 3
0 0 14
```

5. Implemente uma função que receba uma matriz de números inteiros de tamanho $n_l \times n_c$ como parâmetro de entrada e como parâmetro de saída três números inteiros. A função a ser implementada deve armazenar nos parâmetros de saída o maior elemento da matriz e também a linha e coluna onde este está localizado. A função `main` deve ler as dimensões `n_l` e `n_c` da matriz, cada um dos seus elementos e exibir na tela uma mensagem informando o maior elemento da matriz e a linha e coluna onde ele está localizado utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas da matriz:
2
Informe o nr. de colunas da matriz:
4
Informe os elementos da matriz:
8 9 -1 1
2 3 5 1
Maior elemento e 9, esta na linha 0 e coluna 1
```

6. Implemente uma função que receba como parâmetro de entrada uma matriz de números inteiros de tamanho $n_l \times n_c$. A função a ser implementada deve retornar verdadeiro caso a matriz seja uma matriz esparsa ou falso caso contrário. Considere uma matriz $n_l \times n_c$ como sendo esparsa se pelo menos 70% dos seus

elementos forem iguais a zero. A função `main` deve ler as dimensões `nl` e `nc` da matriz, cada um dos seus elementos e exibir na tela uma mensagem informando se a matriz é esparsa ou não. Exemplo:

```
Informe o nr. de linhas da matriz:
2
Informe o nr. de colunas da matriz:
2
Informe os elementos da matriz:
0 1
0 0
A matriz e esparsa
```

7. Implemente uma função que receba uma matriz de números inteiros quadrada de ordem `n` como parâmetro de entrada. A função a ser implementada deve retornar verdadeiro caso a matriz seja uma matriz de permutação ou falso caso contrário. Uma matriz $n \times n$ é de permutação se ela for formada apenas por 0s e 1s e se cada uma de suas linhas e colunas possuir apenas um único elemento igual a 1. A função `main` deve ler a ordem `n` da matriz quadrada, cada um dos seus elementos e exibir na tela uma mensagem informando se a matriz é ou não de permutação utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas/colunas:
4
Informe os elementos da matriz:
0 1 0 0
1 0 0 0
0 0 1 0
0 0 0 1
Matriz informada e de permutacao
```

8. Implemente uma função que receba uma matriz de números inteiros quadrada de ordem `n` como parâmetro de entrada. A função a ser implementada deve retornar verdadeiro caso a matriz seja um quadrado mágico ou falso caso contrário. Uma matriz quadrada é chamada de 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 forem todas iguais. A função `main` deve ler a ordem `n` da matriz quadrada, cada um dos seus elementos e exibir na tela uma mensagem informando se a matriz é ou não um quadrado mágico utilizando a função implementada. Exemplo:

```
Informe o nr. de linhas/colunas:
3
Informe os elementos da matriz:
8 0 7
4 5 6
3 10 2
A matriz informada e um quadrado magico
```

9. Implemente uma função que receba como parâmetro de entrada uma matriz de números inteiros de tamanho `nl` \times `nc` e como parâmetro de saída um vetor de números inteiros. A função a ser implementada deve armazenar em cada posição do vetor o maior elemento de cada coluna da matriz. A função `main` deve ler as dimensões `nl` e `nc` da matriz, cada um dos seus elementos e exibir na tela o vetor resultante. Exemplo:

```
Informe o nr. de linhas da matriz:
3
```

Informe o nr. de colunas da matriz:

3

Informe os elementos da matriz:

5 1 3

4 8 9

2 6 3

Vetor resultante:

5 8 9

10. Implemente uma função que receba como parâmetro de entrada uma matriz 3D de números inteiros de tamanho $np \times nl \times nc$ (número de planos, de linhas e colunas, respectivamente). A função também possui um parâmetro de saída, dado por uma matriz 2D de tamanho $nl \times nc$. A função a ser implementada deve armazenar no parâmetro de saída a matriz 2D composta pela soma de todas as np matrizes de tamanho $nl \times nc$. A função `main` deve ler as dimensões np , nl e nc da matriz 3D, cada um dos seus elementos e exibir na tela a matriz 2D calculada pela função. Exemplo:

Insira as dimensoes da matriz 3D:

3 2 2

Insira a matriz 3D, plano por plano:

0 1

1 0

1 2

2 1

-1 0

0 -1

Soma:

0 3

3 0