



Exercício 01

Programação Aplicada a Sistemas de Energia

Questão 1 (**inversa3x3.m**)

Uma matriz quadrada de determinante não nula tem inversa que pode ser computada por:

$$\mathbf{M}^{-1} = \frac{1}{|\mathbf{M}|} \text{adj}(\mathbf{M}) \quad (1)$$

onde $|\mathbf{M}|$ é a determinante da matriz e $\text{adj}(\mathbf{M})$ sua adjunta.

Para uma matriz 3×3

$$\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (2)$$

o determinante é definido por:

$$|\mathbf{M}| = aei + bfg + cdh - gec - hfa - idb \quad (3)$$

e sua adjunta é definida por:

$$\text{adj}(\mathbf{M}) = \begin{bmatrix} ei-hf & hc-bi & bf-ec \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix} \quad (4)$$

Escreva uma **função** `invM = inversa3x3(M)` que recebe como argumento uma matriz \mathbf{M} de ordem 3, calcula sua inversa através de (1), e retorna a matriz inversa calculada.

Ainda, a função deve verificar possíveis erros como:

- se o argumento não for uma matriz quadrada deve gerar um erro `'Matriz não quadrada'`;
- se a ordem da matriz não for 3 deve gerar um erro `'Matrix de ordem diferente de 3 não suportada'`;
- se o determinante for zero deve gerar um erro `'Matriz de determinante zero não possui inversa'`.

Utilize a função padrão `inv(M)` para validar sua implementação.

† *Embora (1) seja uma formula fechada e válida para matrizes quadradas de qualquer ordem, ela é comutacionalmente ineficiente e os algoritmos utilizados para inversão de matrizes na prática, sobretudo de ordens elevada, são baseadas em escalonamento. A proposta aqui é apenas exercitar a manipulação de elementos em matrizes.*

Questão 2 (**bussola.m**)

Escreva um **script** que ao ser executado pergunta ao usuário um valor de ângulo em graus. Baseado na resposta deste, o programa deve identificar qual ponto cardeal mais próximo que equivale ao ângulo, conforme desenho (ângulo 0° corresponde ao leste), e exibir uma mensagem na forma `'O angulo indicado aponta para o nordeste'`.

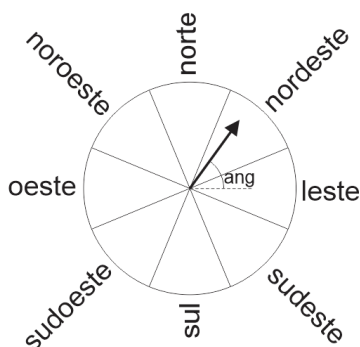
O programa deve suportar ângulos maiores que 360° , bem como ângulos negativos.

Ainda, o algoritmo deve verificar e gerar erros apropriados quando o usuário não digitar um escalar. Por exemplo, ele pode entrar com uma matriz ou como um texto.



Exemplo:

```
1 >> bussola
2 Digite o angulo: 62.18
3 O angulo indicado aponta para o nordeste
```



Questão 3 (multiplo.m)

Escreva uma **função** `multiplo(num, mult)` que recebe dois escalares como argumento e retorna `true` ou `false` para os casos em que `num` é ou não é múltiplo inteiro de `mult`;

Questão 4 (calc_area.m)

A seguir você encontrará algumas tentativas de implementação de um pequeno **script** para o cálculo da área de diferentes formas geométricas. O script pergunta ao usuário qual a largura, qual a altura e qual a forma desejada ('quadrado' e 'triangulo' são suportados). Baseado nas respostas do usuário, o script calcula a área e a apresenta na tela.

Todas as tentativas a seguir apresentam algum erro. Identifique o erro em cada uma delas e corrija-o.

Considere que script está contido em um arquivo `calc_area.m` localizado no diretório corrente de execução do Matlab.

Tentativa 1:

```
1 % inicio do script (calc_area.m)
2 largura = input('Digite a largura em metros: ');
3 altura = input('Digite a altura em metros: ');
4 forma = input('Digite a forma geometrica (quadrado ou triangulo): ', 's');
5
6 if forma == 'quadrado'
7     area = largura * altura;
8 elseif forma == 'triangulo'
9     area = largura * altura / 2;
10 else
11     error(['Forma ', forma, ' nao suportada!']);
12 end
13 disp(['Area calculada do ', forma, ': ', num2str(area), ' m^2']);
14 % fim do script
```



Tentativa 2:

```
1 % inicio do script (calc_area.m)
2 largura = input('Digite a largura em metros: ');
3 altura = input('Digite a altura em metros: ');
4 forma = input('Digite a forma geometrica (quadrado ou triangulo): ', 's');
5
6 if strcmp(forma, 'quadrado')
7     area = largura * altura;
8 else if strcmp(forma, 'triangulo')
9     area = largura * altura / 2;
10 else
11     error(['Forma ', forma, ' nao suportada!']);
12 end
13 disp(['Area calculada do ', forma, ': ', num2str(area), ' m^2']);
14 % fim do script
```

Tentativa 3:

```
1 % inicio do script (calc_area.m)
2 largura = input('Digite a largura em metros: ');
3 altura = input('Digite a altura em metros: ');
4 forma = input('Digite a forma geometrica (quadrado ou triangulo): ', 's');
5
6 if strcmp(forma, 'quadrado')
7     area = largura * altura;
8 elseif strcmp(forma, 'triangulo')
9     area = largura * altura / 2;
10 else
11     error(['Forma ', forma, ' nao suportada!']);
12 end
13 disp(['Area calculada do ', forma, ': ', area, ' m^2']);
14 % fim do script
```

Tentativa 4:

```
1 % inicio do script (calc_area.m)
2 largura = input('Digite a largura em metros: ');
3 altura = input('Digite a altura em metros: ');
4 forma = input('Digite a forma geometrica (quadrado ou triangulo): ', 's');
5
6 if strcmp(forma, 'quadrado')
7     calc_area = largura * altura;
8 elseif strcmp(forma, 'triangulo')
9     calc_area = largura * altura / 2;
10 else
11     error(['Forma ', forma, ' nao suportada!']);
12 end
13 disp(['Area calculada do ', forma, ': ', num2str(calc_area), ' m^2']);
14 % fim do script
```