



## Exercício 02

### Programação Aplicada a Sistemas de Energia

#### Questão 1 (ordena.m)

Escreva uma **função** `[vord, idv]=ordena(v)` que ordena uma lista numérica `v` em ordem crescente. A função recebe como argumento um vetor linha `v` de tamanho arbitrário e deve retornar um vetor linha `vord`, de mesmo tamanho, contendo os valores de `v` ordenados de forma crescente. A função deve ainda retornar um segundo vetor `idv` contendo os índices dos elementos de `v` correspondentes no vetor ordenado `vord`.

Exemplo:

```
1 >> [vord, idv] = ordena([1, 4, 3, 5, 8, 2])
2 vord =
3     [1 2 3 4 5 8]
4
5 idv =
6     [1 6 3 2 4 5]
```

A função do `sort` do matlab faz exatamente a mesma coisa. Utilize a função `sort` do matlab para validar sua implementação para uma gama mais ampla de exemplos.

#### Questão 2 (procura.m)

Escreva uma **função** `[id]=procura(v,p)` que recebe como argumentos uma lista de números `v` e um número `p`. A função deve procurar `p` dentro de `v` e retornar o índice do/dos elementos de `v` que correspondem a `p`, caso contrário, retorna um conjunto vazio.

Exemplo:

```
1 >> procura([2 4 8 1 0 1], 1)
2 ans =
3     [4 6]
```

#### Questão 3 (idsoma.m)

Escreva uma **função** `[idA, idB]=idsoma(v, S)` que identifica um par de elementos de `v` que somados resultam em `S`. A função recebe dois argumentos, o primeiro sendo um vetor linha `v` de tamanho arbitrário e o segundo um número inteiro `S`. A função deve identificar um par de elementos que compõe `v` que somados resultam em `S` e retornar o índice destes elementos dentro do vetor `v`.

Exemplo:

```
1 >> [idA, idB] = idsoma([2, 4, 3, 5, 8], 9)
2 idA =
3     2
4
5 idB =
6     4
```



Caso nenhum par de elementos do vetor  $v$  somados resultem em  $S$ , a função deve retornar  $[\text{NaN}, \text{NaN}]$  ( $\text{NaN} = \text{Not a Number}$ ).

## Questão 4 (**fatorial.m**)

Escreva uma **função** `y=fatorial(x)` que recebe como argumento um número inteiro e retorna o fatorial deste número. Caso o argumento passado à função não seja inteiro ou seja uma matriz a função deve exibir uma mensagem indicando que não pode calcular o fatorial para o argumento recebido.

## Questão 5 (**maxmin.m**)

Escreva uma **função** `[max,min]=maxmin(v)` que recebe como argumento uma lista  $v$  e retorna o valor máximo e mínimo dos elementos desta lista. Se a lista passada como argumento for vazia, retorne `[]` também para `max` e `min`. Se o argumento  $v$  for uma matriz retorne os elementos máximo e mínimo da matriz. Exemplo: `maxmin([2 4 8; 1 0 -1])` deve retornar `[8, -1]`.

Exemplo:

```
1 >> maxmin([2 4 8; 1 0 -1])
2 ans =
3      8     -1
```

## Questão 6 (**arctan.m**)

A função arctangente pode ser aproximada pela série Taylor:

$$\arctan(x) = \sum_{n=1,3,\dots}^{\infty} \frac{x^n}{n} (-1)^{(n+1)/2} \quad (1)$$

Escreva uma **função** que receba  $x$  como argumento e que compute e retorne o valor calculado da série até atingir o primeiro termo em que

$$\left| \frac{x^n}{n} \right| \leq 0.0001 \quad (2)$$

Compare o resultado de sua função com a função padrão `atan(x)`.

## Questão 7 (**desviopadrao.m**)

Escreva uma **função** que recebe como argumento uma lista numérica  $\mathbf{x}$  na forma de uma vetor linha de tamanho arbitrário e retorna o valor calculado do desvio padrao considerando probabilidades iguais para todos os elementos. Ou seja:

$$\sigma(\mathbf{x}) = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x_n - \bar{x})^2} \quad (3)$$

sendo  $\bar{x}$  a média aritmética dos elementos de  $\mathbf{x}$  e  $N$  o número de elementos de  $\mathbf{x}$ .

Compare o resultado de sua função com a função padrão `std(x)`.



## Questão 8 (balanceamento.m)

Escreva uma função que receba como argumento um vetor linha de tamanho arbitrário e retorne o índice do elemento que subdivide a lista em duas partes cujas somas dos elementos é o mais próximo da igualdade. O índice retornado deve corresponder ao primeiro elemento da segunda parte.

Exemplo:

```
1 >> id = balanceamento([2, 4, 1, 6, 3, 5, 4])
2 id =
3     5
```

já que  $2+4+1+6=13$  e  $3+5+4=12$ , sendo 13 e 12 (diferença de 1) é o melhor balanceamento possível para essa lista.

## Questão 9 (encontrarepetidos.m)

Escreva uma função `rep=encontrarepetidos(v1, v2)` que recebe duas listas como argumentos, sendo `v1` e `v2` vetores linhas de tamanho arbitrário e não necessariamente iguais. A função deve criar e retornar uma terceira lista contendo apenas os elementos que estão contidos em ambas as listas `v1` e `v2`.

Exemplo:

```
1 >> rep = encontrarepetidos([1, 4, 2, 6, 9], [8, 7, 2, 0, 1, 6])
2 rep =
3     [1 2 6]
```

## Questão 10 (ordenacolunas.m)

Escreva uma função que recebe como argumento uma matriz de tamanho arbitrário e que retorna uma matrix de mesmo formato, contendo as mesmas colunas da matriz argumento, porém ordenadas de forma crescente com relação ao somatório do elementos de cada coluna.

Exemplo:

```
1 >> ordenacolunas([2 4 6 2
2                   0 9 1 1
3                   7 2 3 0])
4 ans =
5     2 2 6 4
6     1 0 1 9
7     0 7 3 2
```

## Questão 11 (sen.m)

Escreva uma função que recebe como argumento um ângulo e que retorna o seno desse ângulo aproximado pela série

$$\text{sen}(\alpha) = \alpha - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \frac{\alpha^7}{7!} + \dots \quad (4)$$



truncada no primeiro termo em que

$$\left| \frac{\alpha^k}{k!} \right| \leq 0.0001 \quad (5)$$

A série (4) converge para ângulos  $\alpha$  no intervalo  $[-\pi, +\pi]$ . Assim, se o ângulo passado como argumento estiver fora desse intervalo, antes de computar a série, o ângulo deve ser mapeado no intervalo de convergência.