

	<p style="text-align: center;">UNIVERSIDADE FEDERAL DE CAMPINA GRANDE DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO</p>
	Disciplina: CN
	Professor:
	Semestre: Turma: Data: / /

Lista 04: Raizes

Questão 1: Escreva uma função em Python capaz de calcular a aproximação de uma raiz usando o **método da Bissecão**. A sua função deve receber os seguintes parâmetros de entrada:

```
def [aprox, e_rel, e_abs] = bissecao(funcao, xa, xb, int, best_apro)
# funcao      : funcao que se deseja localizar a raiz
# xa          : valor xa
# xb          : valor xb
# int         : número de iterações
# best_apro   : melhor aproximação
# aprox       : raiz encontrada
# e_rel       : erro relativo
# e_abs       : erro absoluto
```

A função deve também imprimir no terminal a seguinte saída de dados (exemplo ilustrativo):

```

Metodo da Bissecao
-----
Int   xa      xb      erro_abs      erro_rel
-----
1     1.0000000000 0.5372626655 0.02988062487314 0.05268620008102
2     0.5372634545 0.5670096854 0.00013360504487 0.00023557546590
..
i     0.5671423445 0.5671432904 0.000000000000014 0.000000000000025
FIM

Funcao_____ : math.exp(-x)-x
Raiz encontrada__ : 0.56714329005687
Erro absoluto____ : 0.000000000000014
Erro relativo____ : 0.000000000000025
```

Finalmente, teste sua função para calcular a aproximação da raiz para as funções:

- $f(x) = 3x^2 + 2x - 2$. Intervalo: $x_1 = 0$ e $x_2 = 1$.
- $f(x) = 2x^3 + x^2 + 2x - 1$. Intervalo: $x_1 = 0$ e $x_2 = 1$.
- $f(x) = x^3 + 2x - 30$. Intervalo: $x_1 = 2$ e $x_2 = 3$
- $f(x) = 4^{-x} - x$. Intervalo: $x_1 = -2$ e $x_2 = 1$.

Questão 2: Escreva uma função em Python capaz de calcular a aproximação de uma raiz usando o **método da Falsa Posição**. A sua função deve receber os seguintes parâmetros de entrada:

```
def [aprox, e_rel, e_abs] = fposicao(funcao, xa, xb, int, best_apro)
# funcao      : funcao que se deseja localizar a raiz
# xa          : valor xa
# xb          : valor xb
# int         : número de iterações
# best_apro   : melhor aproximação
# aprox       : raiz encontrada
# e_rel       : erro relativo
# e_abs       : erro absoluto
```

A função deve também imprimir no terminal a seguinte saída de dados (exemplo ilustrativo):

```

Metodo da Falsa Posicao
-----
Int    xa      xb      erro_abs      erro_rel
-----
1      1.0000000000 0.5372626655 0.02988062487314 0.05268620008102
2      0.5372634545 0.5670096854 0.00013360504487 0.00023557546590
..
i      0.5671423445 0.5671432904 0.000000000000014 0.000000000000025
FIM

Funcao_____ : math.exp(-x)-x
Raiz encontrada__ : 0.56714329005687
Erro absoluto____ : 0.000000000000014
Erro relativo____ : 0.000000000000025
```

Finalmente, teste sua função para calcular a aproximação da raiz para as funções:

- $f(x) = 2x^2 - 4x$. Intervalo: $x_1 = -0.5$ e $x_2 = 0.5$.
- $f(x) = 3x^2 + 2x - 2$. Intervalo $x_1 = 0$ e $x_2 = 1$.
- $f(x) = 2x^3 + x^2 + 2x - 1$. Intervalo: $x_1 = 0$ e $x_2 = 1$.
- $f(x) = x^3 + 2x - 30$. Intervalo: $x_1 = 2$ e $x_2 = 3$.
- $f(x) = 4^{2x^2-5x} - 3$. Intervalo: $x_1 = -0.25$ e $x_2 = 0.15$.

Questão 3: Escreva uma função em Python capaz de calcular a aproximação de uma raiz usando o **método da Secante**. A sua função deve receber os seguintes parâmetros de entrada:

```
def [aprox, e_rel, e_abs] = secante(funcao,xi,ximl,int,best_apro)
# funcao      : funcao que se deseja localizar a raiz
# xi          : valor xi
# ximl        : valor ximl
# int         : número de iterações
# best_apro   : melhor aproximação
# aprox       : raiz encontrada
# e_rel       : erro relativo
# e_abs       : erro absoluto
```

A função deve também imprimir no terminal a seguinte saída de dados (exemplo ilustrativo):

```

Metodo da Secante
-----
Int   xi      ximl      erro_abs      erro_rel
-----
1     1.0000000000 0.5372626655 0.02988062487314 0.05268620008102
2     0.5372634545 0.5670096854 0.00013360504487 0.00023557546590
..
i     0.5671423445 0.5671432904 0.000000000000014 0.000000000000025
FIM

Funcao_____ : math.exp(-x)-x
Raiz encontrada__ : 0.56714329005687
Erro absoluto____ : 0.000000000000014
Erro relativo____ : 0.000000000000025
```

Finalmente, teste sua função para calcular a aproximação da raiz para as funções:

- a) $f(x) = 2^{x^2} - 4x$. Pontos: $x_0 = -0.5$ e $x_1 = -1$.
- b) $f(x) = 4^{x^2} - 6x$. Pontos: $x_0 = 1.5$ e $x_1 = 2$.

Questão 4: Escreva uma função em Python capaz de calcular a aproximação de uma raiz usando o **método da Secante Modificado**. A sua função deve receber os seguintes parâmetros de entrada:

```
def [aprox, e_rel, e_abs] = secantemod(funcao, xi, d, int, best_apro)
# funcao      : funcao que se deseja localizar a raiz
# xi          : valor xi
# d           : valor delta
# int         : número de iterações
# best_apro   : melhor aproximação
# aprox       : raiz encontrada
# e_rel       : erro relativo
# e_abs       : erro absoluto
```

A função deve também imprimir no terminal a seguinte saída de dados (exemplo ilustrativo):

```

Metodo da Secante Modificado
-----
Int    xi          erro_abs      erro_rel
-----
1      1.0000000000 0.02988062487314 0.05268620008102
2      0.5372634545 0.00013360504487 0.00023557546590
..
i      0.5671423445 0.000000000000014 0.000000000000025
FIM

Funcao_____ : math.exp(-x)-x
Raiz encontrada__ : 0.56714329005687
Erro absoluto____ : 0.000000000000014
Erro relativo____ : 0.000000000000025
```

Finalmente, teste sua função para calcular a aproximação da raiz para as funções:

- a) $f(x) = 2^{x^2} - 4x$. Ponto: $x_i = -0.5$
- b) $f(x) = 4^{x^2} - 6x$. Ponto: $x_i = 1.5$