AGA0503 - Exercício de Programação 1
Primeiro semestre de 2025

Devolução: 29/09 (Atenção: desconto de 0,5 ponto por dia de atraso)

1) **Ordenamento de um conjunto de dados (3 pontos + 10% caso use OOP)**

Dois procedimentos frequentemente empregados em técnicas numéricas são o de organizar um conjunto de dados (ex: colocar uma tabela numérica em ordem numérica crescente) e selecionar partes de um conjunto de números de acordo com alguma regra (ex: encontrar a posição em que um número se encontra na tabela).

Para ordenamento de dados, um dos métodos mais simples é o método da inserção direta. A melhor forma de descrever esse método é fazer uma analogia com a maneira com a qual um jogador de cartas organiza o maço de cartas em sua mão:

- pega a primeira carta;
- pega a segunda carta e a insere em ordem com respeito à primeira;
- pega a terceira carta e a insere em sequência com respeito às duas primeiras;
- etc.

O objetivo desde exercício é fazer uma subrotina que, dado um array numérico de dimensão arbitrária *N,* organizado aleatóriamente, retorne um array ordenado em ordem crescente. Além disso, a subrotina deve contar o número de passos necessários para se cumprir a tarefa (*N*passos). Por passo, compreende-se o número de operações de memória feito.

Para gerar um array com *N* números ordenados de forma aleatória use a subrotina intrínseca do Fortran RANDOM_NUMBER, que é chamada da seguinte forma:

```
CALL RANDOM_NUMBER(r1)
```

onde r1 é uma variável real.

Faça uma tabela do número de passos necessários para se ordernar os dados para valores de *N* entre 10 e 100, em passos de 5. A partir dessa tabela determine a dependência funcional de N_{passos} com *N*.

Entregar caderno no Collab com:

a) código fonte da subrotina e do programa que a chama
b) Exemplo de funcionamento do programa, mostrando um array aleatório com 20 números e o array ordenado.
c) Gráfico em Python da tabela acima, com a determinação da dependência funcional de N_{passos} com *N*.
d) Texto discutindo seus resultados

2) **Zero de Funções (3 pontos + 10% caso use OOP)**

Implemente uma subrotina ou função em Fortran, C++ ou Python para encontrar o zero de uma função genérica, utilizando dois métodos numéricos clássicos:

- Método da Falsa Posição
- Método da Dicotomia (Bisseção)

O usuário deve poder escolher qual método deseja utilizar.

Especificações:

Sua função/subrotina deve:

1. Receber como **entrada**:
    - A **função matemática** cuja raiz se deseja encontrar;
    - O **intervalo inicial** *[a, b]* que contém a raiz (com verificação pelo Teorema de Bolzano)
    - A **tolerância relativa** (precisão desejada para o resultado);
    - O **número máximo de iterações** (opcional)
2. Retornar como **saída**:
    - O valor aproximado da raiz;
    - O número de iterações realizadas;
    - O erro estimado final.

Requisitos de implementação:

- **Organização do código**: nomes de variáveis intuitivos, consistência nos tipos, clareza na lógica.
- **Documentação**: comentários explicativos no código (o que cada parte faz, não apenas "como").
- **Modularização**: separar a implementação dos métodos (por exemplo, funções distintas para dicotomia e falsa posição).

Entregar: caderno do Collab com o código da função.

3) **Teste da Rotina do Item 2 (1 ponto)**

Teste a rotina do item 2 para encontrar a raiz da função $f(x)=\cos(x)-x$, cujo resultado é bem conhecido. Teste ambos os métodos implementados.

Entregar: caderno do collab com o teste (pode ser o mesmo anterior) e um texto discutindo os resultados (por exemplo, velocidade de convergência)

4) Considere o problema descrito na seção 4.2.3 da apostila. (**3 pontos**)

   a. Implemente uma função em Fortran, C ou Python que calcule a função $f(n)$ da equação 4.6. Dicas: 1) programe antes uma função que calcule o valor da função de corpo negro, 2) Em Fortran, use módulos para te ajudar a organizar os dados.
   b. Use esta função e a rotina do item 2 para achar o valor da densidade numérica de grãos, dados:

      $R = 20\ Rsol$
      $T_{ef} = 3000$ K
      $R_i = 300$ Rsol
      $R_e = 1500$ Rsol
      $a = 0{,}20$ μm
      $T_{\text{poeira}} = 1000$K
      $J - K = 0{,}5$.

Use uma **precisão relativa** ε = 0,00001 para o critério de convergência do método.

Entregar:

1) Link de caderno no Collab com a implementação do item a)
2) Texto curto mostrando sua resposta.