**AGA0503 - Exercício de Programação 1**
**Primeiro semestre de 2022**

Devolução: 29/04 (Atenção: desconto de 0,5 ponto por dia de atraso)

## 1) Ordenamento de um conjunto de dados (2 pontos)

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).

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 link de 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) Representação **(1 ponto)**

Considere um sistema de ponto-flutuante com 12 bits na mantissa.

a. Escreva o número $x_1$ = -.111010101001 x $2^6$ no sistema decimal.
b. Neste sistema, qual o valor de $\varepsilon_m$?
c. O que é um número denormalizado? Porque isto é sério do ponto de vista computacional? Como fazer para evitar que tal situação ocorra?

3) Resolva numericamente (com a calculadora) a equação $\log_{10}(x^2-1) = 0$ usando o método da bissecção. Use $\varepsilon_m$ = 0.0001. **(1 ponto)**

4) Repita o exercício 3 com o método de Newton-Raphson. **(1 ponto)**

5) Repita o exercício 3 com o método das secantes. Qual dos métodos converge mais rapidamente? Discuta **(1 ponto)**

**Entregar (exercícios 2 a 5):** papel escrito à mão com os resultados de todas as iterações feitas

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

a. Implemente uma função em Fortran, C ou Python que calcule a função $f(n)$ da equação 4.6. Dica: programe antes uma função que calcule o valor da função de corpo negro
b. Implemente uma subrotina ou função em Fortran, C ou Python que ache o zero de uma função genérica usando o **método da falsa posição ou método da dicotomia**
c. Use a e b para achar o valor da densidade numérica de grãos, dados:

$R$ = 20 *Rsol*
$T_{ef}$ = 3500 K
$R_i$ = 100 Rsol
$R_e$ = 2000 Rsol
$a$ = 0,20 $\mu$m
$T_{\text{poeira}}$ = 900K
$J - K$ = 0,6.

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

**Entregar link de caderno no Collab com as funções a e b implementadas, e programa que as chama e mostra o resultado.**