

1. Criar uma função `determinarTriangulo()`, recebendo três valores float e retorne uma string com o nome de qual triângulo seria (Equilátero, Isósceles ou escaleno).

Realize três chamadas a esta função, passando valores para testar esta função em situações que retornem os três tipos de triângulos.

Observação: Para formar os lados de um triângulo cada um dos valores deve ser menor que a soma dos outros dois. Um triângulo pode ser : equilátero (3 lados iguais), isósceles (apenas 2 lados iguais) e escaleno (3 lados diferentes).

2. Crie um função `situacao()`, que recebe o nome, as três notas e o número de faltas de um aluno e calcule e retorne uma string com sua situação ("Aprovado", "Reprovado por Falta" ou "Reprovado por média").

Faça três chamadas a esta função, passando valores para testá-lo nestas três situações.

Observação: A média para aprovação é 5,0 (cinco) e o limite de faltas é 18 e a reprovação por falta sobrepõe a reprovação por média.

3. Faça um algoritmo que receba 10 números inteiros do usuário e armazene-os em um vetor chamado `numeros`. Valide-os da seguinte maneira: Não serão aceitos números repetidos, ou seja, proíba o usuário de informar números já informados anteriormente. 10 números distintos devem ser informados, armazenados e apresentados ao final do algoritmo.

4. Faça um algoritmo que crie e preencha dois vetores (A e B) de dez elementos numéricos cada um. Crie um terceiro vetor (C) de 20 posições e preencha-o intercalando com os números presentes em A e B.

VETOR A

1	2	3	4	5	6	7	8	9	10
3	5	4	2	2	5	3	2	5	9

VETOR B

1	2	3	4	5	6	7	8	9	10
7	15	20	0	18	4	55	23	8	6

VETOR C

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	7	5	15	4	20	2	0	2	18	5	4	3	55	2	23	5	8	9	6

5. Faça um algoritmo que receba 10 números inteiros quaisquer informados pelo usuário, armazenando-os um vetor chamado `numeros`. Coloque-os em ordem crescente no MESMO vetor, e exiba-os em tela.