



**Atividade Avaliativa #01:** Comandos Condicionais

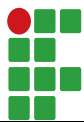
**Observações:**

1. Os programas deverão ser desenvolvidos em linguagem PYTHON;
2. Cada questão deverá ser respondida em arquivos em separado;
3. As respostas deverão ser submetidas no link correspondente a essa lista disponível no Moodle;
4. Atentem para o prazo de submissão. Não serão aceitos envios posteriores a data limite.

1. **(Valor: 10 pontos)** Faça um programa que aceita um número de até 4 dígitos (0 a 9999) e exiba a soma dos seus algarismos.
2. **(Valor: 10 pontos)** Faça um programa que solicite ao usuário um valor decimal positivo (esse valor corresponde ao valor de um saque em um terminal de caixa eletrônico) e que calcule a quantidade de cédulas de R\$ 100,00, R\$ 50,00, R\$ 20,00, R\$ 10,00, R\$ 5,00 e R\$ 2,00 e de moedas de R\$ 1,00, R\$ 0,50, R\$ 0,25, R\$ 0,10, R\$ 0,05 e R\$ 0,01.
3. **(Valor: 10 pontos)** Uma família fez uma viagem de carro e quer detalhes sobre o desempenho do veículo. Faça um programa que pergunta: o momento inicial da partida (hora e minuto), o momento de chegada (hora e minuto), o número de segundos parados para descanso, o número de litros de combustível gasto (em l), o preço do litro de combustível (em R\$) e a distância percorrida (em Km);

Após receber os dados, o programa informa dados típicos de um computador de bordo:

- a) o tempo de viagem (em segundos)
- b) a velocidade média (Km/h) global e a velocidade média em movimento (Km/h)
- c) o custo da viagem com combustível (em R\$)
- d) o desempenho do carro (em Km/l, l/h e R\$/Km).



4. (Valor: 10 pontos) O número de dias decorridos entre duas datas é importante em uma infinidade de situações da vida cotidiana. Faça um programa que receba inicialmente dois valores (dia inicial e mês inicial), depois mais dois valores (dia final, mês final), ao final deve indicar quantos dias se passaram entre as datas (considere que o ano não é bissexto).

Exemplos:

- 02 de 05 até 03 de 05 - 1 dia
- 27 de 04 até 03 de 05 - 6 dias
- 03 de 02 até 03 de 05 - 79 dias

Não esqueça de verificar se a data inicial é menor ou igual à data final.

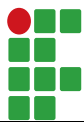
**Dica:** conte o número de dias até cada uma das datas e subtraia esses números.

5. (Valor: 15 pontos) Em um estacionamento, as tarifas são as seguintes (cumulativas):

- Até duas horas: R\$ 8,00/hora ou fração;
- Entre três e quatro horas: R\$ 5,00/hora ou fração;
- Horas seguintes: R\$ 3,00/hora ou fração.
- Depois de 12h, o custo passa a ser fixo: R\$ 30,00

Faça um programa que leia o número de minutos que um veículo permaneceu no estacionamento e exiba o valor a ser pago pelo responsável.

Como exemplo, considere que o carro ficou 310 minutos no estacionamento; deve pagar: R\$ 16,00 (pelas duas primeiras horas) + R\$ 10,00 (pela terceira e quarta horas) + R\$ 6,00 (pela quinta hora e fração da sexta hora): total de R\$ 32,00



6. (Valor: 20 pontos) Com base na nova legislação da Previdência Brasileira, regulamentada pela Lei Complementar nº 1354/2020 e pela Emenda à Constituição nº 49/2020, desenvolva um programa em Python que solicite as seguintes informações de uma pessoa:

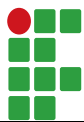
- Sexo da pessoa (masculino/feminino).
- Data de nascimento da pessoa (no formato DD/MM/AAAA).
- Data de início da contribuição previdenciária da pessoa (no formato DD/MM/AAAA).

O programa deve então calcular e exibir a data em que a pessoa poderá se aposentar, considerando as seguintes regras:

- Aposentadoria por Idade:
  - Homens podem se aposentar aos 65 anos de idade.
  - Mulheres podem se aposentar aos 62 anos de idade.
  - É necessário ter pelo menos 15 anos de contribuição.
- Aposentadoria por Tempo de Contribuição:
  - Homens podem se aposentar após 35 anos de contribuição.
  - Mulheres podem se aposentar após 30 anos de contribuição.

Implemente o programa em Python para calcular a data de aposentadoria e exibi-la como resultado.

**Dica:** Utilize as bibliotecas **datetime** e **dateutil**.



7. (Valor: 25 pontos) Desenvolva um código em Python que solicite ao usuário que informe os comprimentos dos três lados de um triângulo. Em seguida, verifique se esses comprimentos podem formar um triângulo. Caso afirmativo, calcule e informe os valores dos ângulos do triângulo e classifique-o quanto aos lados e aos ângulos.

- Instruções:
  - a) Peça ao usuário para inserir os comprimentos dos três lados do triângulo.
  - b) Verifique se os comprimentos fornecidos podem formar um triângulo. Caso contrário, informe que não é possível formar um triângulo com esses lados.
  - c) Se for possível formar um triângulo, calcule os três ângulos do triângulo.
  - d) Classifique o triângulo quanto aos lados (equilátero, isósceles ou escaleno) e aos ângulos (agudo, obtuso ou retângulo).
  - e) Exiba a classificação do triângulo quanto aos lados e aos ângulos.
- Observações:
  - Para determinar se os lados fornecidos pelo usuário podem formar um triângulo, é necessário verificar a seguinte condição: A soma de quaisquer dois lados de um triângulo deve ser sempre maior que o terceiro lado.
  - Você pode usar a Lei dos cossenos para calcular os ângulos de um triângulo.
  - Para classificar quanto aos lados, verifique se os três lados são iguais (equilátero), se dois lados são iguais (isósceles) ou se todos os lados são diferentes (escaleno).
  - Para classificar quanto aos ângulos, verifique se um dos ângulos é maior que 90 graus (obtuso), se um dos ângulos é igual a 90 graus (retângulo) ou se todos os ângulos são menores que 90 graus (agudo).
  - Considere que os ângulos são expressos em graus.

Desenvolva o código em Python para atender às especificações acima.

**Dica:** Utilize a biblioteca **math**.