

## Avaliação Prática 3

Valor total: **10 pontos**

---

Adriano César Machado Pereira  
João Guilherme Maia de Menezes

### Regras de Conduta

- Esta é uma atividade avaliativa que deve ser realizada de maneira individual e sem consulta.
- Celulares devem permanecer desligados durante a atividade.
- Espera-se que o aluno realize suas atividades com honestidade e integridade.
- Falhas de conduta, como cópia de atividades de colegas, serão punidas com dedução parcial ou total da nota, além de outras penalidades, segundo as normas do Colegiado de Graduação.

### Instruções para Submissão

Na avaliação prática de hoje, você terá que elaborar programas para resolver problemas diversos, conforme descrito abaixo. Cada uma das soluções deverá ser implementada em seu próprio arquivo com extensão `.py`. Por exemplo, a solução para o problema 1 deverá ser implementada em um arquivo chamado `problema1.py`, a solução para o problema 2 deverá ser implementada no arquivo `problema2.py` e assim por diante. Finalmente, submeta cada um dos arquivos pelo Moodle.

**Dica:** se você tiver problemas com caracteres especiais (caracteres com acentos, por exemplo), adicione a linha abaixo na primeira linha de todos os arquivos `*.py`

```
# -*- coding: utf-8 -*-
```

### Problema 1 - (3 pontos)

Uma Progressão Aritmética (PA) é uma sequência de números onde a diferença entre dois termos consecutivos é sempre a mesma, e essa diferença é chamada de razão ( $r$ ). Já uma Progressão Geométrica

(PG) corresponde a uma sequência numérica cujo quociente ( $q$ ) ou razão entre um número e outro (exceto o primeiro) é sempre igual.

Escreva uma função chamada `progressoes` que recebe como **parâmetros** o termo inicial ( $a_1$ ), a razão e a quantidade de termos ( $n$ ). Em seguida, a função deve calcular o  $n$ ésimo termo ( $a_n$ ) da Progressão Aritmética (PA) e da Progressão Geométrica (PG). As fórmulas para o cálculo do  $n$ ésimo termo da PA e da PG são, respectivamente:

$$a_n = a_1 + (n - 1) \times r \quad (0.1)$$

$$a_n = a_1 \times q^{(n-1)} \quad (0.2)$$

Finalmente, se o  $n$ ésimo termo da PA for maior que o da PG, a função deve **retornar** a soma ( $S_n$ ) dos  $n$  primeiros termos da PA. Caso contrário, a função deve retornar a soma ( $S_n$ ) dos  $n$  primeiros termos da PG. As fórmulas da soma dos  $n$  primeiros termos de uma PA e de uma PG são, respectivamente:

$$S_n = n \times \frac{(a_1 + a_n)}{2} \quad (0.3)$$

$$S_n = a_1 \times \frac{(q^n - 1)}{q - 1} \quad (0.4)$$

**Observação 1:** O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema1.py`.

**Observação 2:** Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

#### Exemplo 1 de execução do programa:

Digite o termo inicial: 0  
Digite a razão: 2  
Digite a quantidade de termos: 10  
**Resultado: 90**

#### Exemplo 2 de execução do programa:

Digite o termo inicial: 1  
Digite a razão: 1  
Digite a quantidade de termos: 100  
**Resultado: 5050**

#### Exemplo 3 de execução do programa:

Digite o termo inicial: 3  
Digite a razão: 3  
Digite a quantidade de termos: 3  
**Resultado: 39**

## Problema 2 - (2 pontos)

Escreva uma função chamada `consumo` que recebe como **parâmetros** a distância percorrida em quilômetros e a quantidade de litros de gasolina consumidos por um veículo para percorrer tal distância. Sua função deve **retornar** uma mensagem de acordo com a tabela abaixo:

Consumo	Km/l	Mensagem
Menor que	8	Venda o carro!
Entre	8 e 12	Econômico!
Maior que	12	Super econômico!

**Observação 1:** O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema2.py`.

**Observação 2:** Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

### Exemplo 1 de execução do programa:

Digite a distância: **100**

Digite a quantidade de gasolina consumida: **10**

**Econômico!**

### Exemplo 2 de execução do programa:

Digite a distância: **30**

Digite a quantidade de gasolina consumida: **4.5**

**Venda o carro!**

## Problema 3 - (2 pontos)

Escreva uma função chamada `estacionamento` que recebe como **parâmetros** a hora e minuto de entrada, e hora e minuto de saída de um estacionamento e **retorna** o valor total devido de acordo com as seguintes regras:

Quantidade de Horas	Tarifa
Até duas horas	R\$ 1.00 para cada hora
Entre três e quatro horas	R\$ 1.40 para cada hora
Acima de quatro horas	R\$ 2.00 para cada hora

**O número de horas a pagar é sempre inteiro e arredondado por excesso.** Por exemplo, quem estacionar por 1 minuto pagará por 1 hora. Quem estacionar durante 61 minutos pagará por 2 horas. Perceba que os momentos de entrada e saída do estacionamento são apresentados na forma de pares de inteiros, representando horas e minutos. Por exemplo, o par (12, 50) representará "dez para a uma da tarde". Admite-se que a entrada e a saída se dão com intervalo não superior a 24 horas. Se o horário de entrada for igual ao horário de saída, então o tempo de permanência é de 24 horas.

**Observação 1:** O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema3.py`.

**Observação 2:** Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

**Exemplo 1 de execução do programa:**

Digite a hora de chegada: **18**  
Digite o minuto de chegada: **50**  
Digite a hora da partida: **22**  
Digite o minuto da partida: **49**  
Preço: R\$ **5.60**

**Exemplo 2 de execução do programa:**

Digite a hora de chegada: **20**  
Digite o minuto de chegada: **30**  
Digite a hora da partida: **8**  
Digite o minuto da partida: **00**  
Preço: R\$ **24.00**

## Problema 4 - (3 pontos)

Escreva uma função chamada `valor_energia` que recebe como **parâmetros** a quantidade de kWh consumida no mês e o tipo de instalação (R para residências, C para comércios e I para indústrias). A função deve **retornar** o preço a pagar pelo fornecimento de energia elétrica, conforme as regras definidas na tabela a seguir:

Preço por tipo e faixa de consumo		
Tipo	Faixa(kWh)	Preço do kWh
Residencial	Até 500	R\$ 0,40
	Acima de 500	R\$ 0,65
Comercial	Até 1000	R\$ 0,55
	Acima de 1000	R\$ 0,60
Industrial	Até 5000	R\$ 0,55
	Acima de 5000	R\$ 0,60

**Observação 1:** O nome da função deve ser exatamente como especificado acima e seu arquivo deve obrigatoriamente se chamar `problema4.py`.

**Observação 2:** Seu programa deve conter apenas a função descrita acima. Você não precisa realizar a entrada e saída de dados (não precisa usar as funções `input()` e `print()`).

**Exemplo 1 de execução do programa:**

Digite a quantidade de kWh consumida: **580**  
Digite o tipo de instalação: **R**

Preço a pagar: **377.00**

**Exemplo 2 de execução do programa:**

Digite a quantidade de kWh consumida: **4200**

Digite o tipo de instalação: **I**

Preço a pagar: **2310.00**