

### QUESTÃO 1 de 4 - Conteúdo até Aula 03

Enunciado: Imagina-se que você é um dos programadores responsáveis pela construção de app para uma empresa X que vende Planos de Saúde. Uma das estratégias dessa empresa X é cobrar um valor diferente com base na idade do cliente, conforme a **listagem abaixo**:

- Se a idade for **maior ou igual** que **0** e **menor** que **19**, o valor será de **100%** do **valor base** do plano (100 / 100);
- Se a idade for **maior ou igual** que **19** e **menor** que **29**, o valor será de **150%** do **valor base** do plano (150 / 100);
- Se a idade for **maior ou igual** que **29** e **menor** que **39**, o valor será de **225%** do **valor base** do plano (225 / 100);
- Se a idade for **maior ou igual** que **39** e **menor** que **49**, o valor será de **240%** do **valor base** do plano (240 / 100);
- Se a idade for **maior ou igual** que **49** e **menor** que **59**, o valor será de **350%** do **valor base** do plano (350 / 100);
- Se a idade for **maior ou igual** que **59**, o valor será de **600%** do **valor base** do plano (600 / 100);

O valor mensal do plano é calculado da seguinte maneira:

$$\text{valorMensal} = \text{valorBase} * \text{porcentagem}$$


**Exemplo:** Se o **valorBase** informado for **100.00** e a **idade** for **45** anos (**240%** segundo a tabela acima)

$$\text{valorMensal} = 100.00 * \left(\frac{240}{100}\right) = R\$ 240.00$$

Elabore um programa em Python que:

- Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).  
Por exemplo: **print("Sistema desenvolvido por Bruno Kostiuk")** [EXIGÊNCIA DE CÓDIGO 1 de 6];
- Deve-se implementar o input do **valorBase** do plano e da **idade** do cliente [EXIGÊNCIA DE CÓDIGO 2 de 6];
- Deve-se implementar as regras de valores **conforme a enunciado acima** (obs.: atente-se as condições de menor, igual e maior) [EXIGÊNCIA DE CÓDIGO 3 de 6];
- Deve-se implementar o **valorMensal** [EXIGÊNCIA DE CÓDIGO 4 de 6];
- Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
- Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
- Deve-se apresentar na saída de console uma mensagem com seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];
- Deve-se apresentar na saída de console a utilização do sistema informando uma **idade maior ou igual a 29 anos**, apresentando na saída de console o **valorMensal** do plano [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2];

## EXEMPLO DE SAÍDA DE CONSOLE:



```
Bem vindo ao Sistema do Bruno KostiuK
Informe o valor Base do plano: R$ 134.05
Informe a idade do cliente: 34
O valor mensal do plano é de: R$ 301.61
```

Figura 1.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o valorBase do plano (pode ser qualquer valor) e a idade (maior ou igual a 29 anos **EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2**), e é apresentado o valorMensal.

### ***Apresentação de Código da Questão 1:***

```
print('Sistema desenvolvido por Olavo Molina')

# Entradas a serem feitas pelo usuário para inserir os valores base do plano e idade.
valorBase = float(input('Informe o valor base do plano:'))
idade = int(input('Informe a idade do cliente:'))

# Condicionantes e porcentagens para cada faixa etária a serem usadas nos cálculos.
if (0 <= idade < 19) :
    porcentagem = 100 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')
elif (19 <= idade < 29) :
    porcentagem = 150 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')
elif (29 <= idade < 39) :
    porcentagem = 225 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')
elif (39 <= idade < 49) :
    porcentagem = 240 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')
elif (49 <= idade < 59) :
    porcentagem = 350 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')
elif (idade >= 59) :
    porcentagem = 600 / 100
    valor = valorBase * porcentagem
    print(f'O valor mensal do plano é: R$ {valor:.2f}')

# Retorno se uma idade estiver fora das faixas.
else:
    print('Idade inválida')
```

*Apresentação de Saída do Console da Questão 1:*



```
Sistema desenvolvido por Olavo Molina  
Informe o valor base do plano: 300  
Informe a idade do cliente:31  
O valor mensal do plano é: R$ 675.00
```

## QUESTÃO 2 de 4 - Conteúdo até aula 04

Enunciado: Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma Pizzaria que vende sabores de Pizzas Doces e Pizzas Salgadas. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.

A Loja possui seguinte relação:

- Tamanho **P**: Pizza Salgada (**PS**) custa 30 reais e a Pizza Doce (**PD**) custa 34 reais;
- Tamanho **M**: Pizza Salgada (**PS**) custa 45 reais e a Pizza Doce (**PD**) custa 48 reais;
- Tamanho **G**: Pizza Salgada (**PS**) custa 60 reais e a Pizza Doce (**PD**) custa 66 reais;

Elabore um programa em Python que:

- A. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print("Bem-vindos a Pizzaria do Bruno Kostiuk")**

Além do seu nome completo, deve-se implementar um **print com um Menu** para o cliente.

**[EXIGÊNCIA DE CÓDIGO 1 de 8];**

- B. Deve-se implementar o input do **sabor** (PS/PD) e o print "Sabor inválido. Tente novamente" se o usuário entra com valor diferente de PS e PD **[EXIGÊNCIA DE CÓDIGO 2 de 8];**
- C. Deve-se implementar o input do **tamanho** (P/M/G) e o print "Tamanho inválido. Tente novamente" se o usuário com entra valor diferente de P, M ou G **[EXIGÊNCIA DE CÓDIGO 3 de 8];**
- D. Deve-se implementar **if, elif e/ou else**, utilizando o modelo **aninhado** (aula 3 – Tema 4) com cada uma das combinações de **sabor e tamanho** **[EXIGÊNCIA DE CÓDIGO 4 de 8];**
- E. Deve-se implementar um **acumulador** para somar os valores dos pedidos (valor total do pedido) **[EXIGÊNCIA DE CÓDIGO 5 de 8];**
- F. Deve-se implementar o input com a pergunta: "Deseja pedir mais alguma coisa?". Se sim **repetir a partir do item B**, senão encerrar o programa executar o print do **acumulador** **[EXIGÊNCIA DE CÓDIGO 6 de 8];**
- G. Deve-se implementar as estruturas de **while, break, continue (todas elas)** **[EXIGÊNCIA DE CÓDIGO 7 de 8];**
- H. Deve-se inserir comentários **relevantes** no código **[EXIGÊNCIA DE CÓDIGO 8 de 8];**
- I. Deve-se apresentar na saída de console uma mensagem com o seu nome completo e o menu para o cliente conhecer as opções **[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];**
- J. Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];**
- K. Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];**
- L. Deve-se apresentar na saída de console um pedido com duas opções sabores diferentes e com tamanhos diferentes **[EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];**

## EXEMPLO DE SAÍDA DE CONSOLE:

```
➡ ----- Bem-vindo a Pizzaria do Bruno Kostiuk -----  
-----Cardápio-----  
-----  
---| Tamanho | Pizza Salgada(PS) | Pizza Doce(PD) | ---  
---| P | R$ 30.00 | R$ 34.00 | ---  
---| M | R$ 45.00 | R$ 48.00 | ---  
---| G | R$ 60.00 | R$ 66.00 | ---  
-----  
Entre com o sabor desejado (PS/PD): PG  
Sabor inválido. Tente novamente  
  
Entre com o sabor desejado (PS/PD): PS  
Entre com o tamanho desejado (P/M/G): EXGG  
Tamanho inválido. Tente novamente  
  
Entre com o sabor desejado (PS/PD): PS  
Entre com o tamanho desejado (P/M/G): G  
Você pediu uma Pizza Salgada no tamanho G: R$ 60.00  
  
Deseja mais alguma coisa? (S/N): S  
Entre com o sabor desejado (PS/PD): PD  
Entre com o tamanho desejado (P/M/G): M  
Você pediu uma Pizza Doce no tamanho M: R$ 48.00  
  
Deseja mais alguma coisa? (S/N): N  
  
O valor total a ser pago: R$ 108.00
```

Mensagem com seu nome completo e Menu de opções (cardápio)

Usuário errou o sabor

Usuário errou o tamanho

Pedido com 2 itens de tamanho e sabores diferentes

**Figura 2.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o sabor e o tamanho. Há uma tentativa de pedido que se errou o sabor e outra que se errou o tamanho. Há também um pedido com dois itens com sabores e tamanhos diferentes.**

## ***Apresentação de Código da Questão 2:***

```
print ( '|','-' * 56, '|' )
print ( '|', ' ' * 8, 'Bem-vindos a Pizzaria do Olavo Molina', ' ' * 9, '|' )
print ( '|','-' * 56, '|' )
print ( '|','-' * 20, 'Cardápio', '-'* 26, '|')
print ( '|','-' * 2, 'Pizza Salgada (P = R$ 30 / M = R$ 45 / G = R$ 60)', '-'* 2, '|')
print ( '|','-' * 2, 'Pizza Doce', ' '* 2, '(P = R$ 34 / M = R$ 48 / G = R$ 66)', '-'* 2, '|')
print ( '|','-' * 56, '|' )

# Totalizador zerado para cálculo de total de pedido
total_pedido = 0

while True:
# Entrada de sabor da pizza pelo usuário.
    sabor = input('Qual o sabor de Pizza desejada:(PS (Salgada) / PD (Doce)):')
    if sabor not in ['PS','PD']:
        print('Sabor inválido, tente novamente."\n')
        continue
# Entrada de tamaho da pizza pelo usuário.
    tamanho = input('Entre qual tamanho da pizza:(P/M/G):')
    if tamanho not in ['P','M','G']:
        print('Tamanho inválido, tente novamente."\n')
        continue
# Combinação de sabor e tamanho com valores para cálculo de total do pedido
    if sabor == 'PS':
        if tamanho == 'P':
            preco = 30
        if tamanho == 'M':
            preco = 45
        if tamanho == 'G':
            preco = 60
    elif sabor == 'PD':
        if tamanho == 'P':
            preco = 34
        if tamanho == 'M':
            preco = 48
        if tamanho == 'G':
            preco = 66
    if sabor == 'PS':
        nome_sabor = "Pizza Salgada"
    elif sabor == 'PD':
        nome_sabor = "Pizza Doce"

    print(f'Você escolheu uma {nome_sabor} do tamanho {tamanho}. Preço: R$ {preco:.2f}."\n')

# Acumulador para cálculo de total do pedido
```

```

total_pedido = total_pedido + preco
# Verificação se o cliente quer mais alguma pizza
outro_pedido = input('Deseja pedir mais alguma coisa? (S para sim / N para não):')
if outro_pedido == "N":
    break
# Exibição do valor total do pedido
print('\n\nValor total do seu pedido: R$ {total_pedido:.2f}')

```

### Apresentação de Saída do Console da Questão 2:



```

-----
| Bem-vindos a Pizzaria do Olavo Molina |
|-----|
|----- Cardápio -----|
| -- Pizza Salgada (P = R$ 30 / M = R$ 45 / G = R$ 60) -- |
| -- Pizza Doce (P = R$ 34 / M = R$ 48 / G = R$ 66) -- |
|-----|
Qual o sabor de Pizza desejada:(PS (Salgada) / PD (Doce)):PGG
Sabor inválido, tente novamente.

Qual o sabor de Pizza desejada:(PS (Salgada) / PD (Doce)):PS
Entre qual tamanho da pizza:(P/M/G):GGG
Tamanho inválido, tente novamente.

Qual o sabor de Pizza desejada:(PS (Salgada) / PD (Doce)):PS
Entre qual tamanho da pizza:(P/M/G):G
Você escolheu uma Pizza Salgada do tamanho G. Preço: R$ 60.00.

Deseja pedir mais alguma coisa? (S para sim / N para não):S
Qual o sabor de Pizza desejada:(PS (Salgada) / PD (Doce)):PD
Entre qual tamanho da pizza:(P/M/G):M
Você escolheu uma Pizza Doce do tamanho M. Preço: R$ 48.00.

Deseja pedir mais alguma coisa? (S para sim / N para não):N

Valor total do seu pedido: R$ 108.00

```



### QUESTÃO 3 de 4 - Conteúdo até aula 05

Enunciado: Você foi contratado para desenvolver um sistema de Venda de uma Empresa Y que vende toras de árvore para outras empresas que vendem madeira. Você ficou com a parte de desenvolver a interface com o cliente.

A Empresa Y opera as vendas da seguinte maneira:

- Tora de Pinho (PIN), o valor do metro cúbico (m³) é de cento e cinquenta reais e quarenta centavos;
- Tora de Peroba (PER), o valor do metro cúbico (m³) é de cento e setenta reais e vinte centavos;
- Tora de Mogno (MOG), o valor do metro cúbico (m³) é de cento e noventa reais e noventa centavos;
- Tora de Ipê (IPE), o valor do metro cúbico (m³) é de duzentos e dez reais e dez centavos;
- Tora de Imbuia (IMB), o valor do metro cúbico (m³) é de duzentos e vinte reais e setenta centavos;
  
- Se a quantidade (em m³) de toras for **menor** que 100 não há desconto na venda (0/100);
- Se a quantidade (em m³) de toras for **igual ou maior** que 100 e **menor** que 500, o desconto será de 4% (4/100);
- Se a quantidade (em m³) de toras for **igual ou maior** que 500 e **menor** que 1000, o desconto será de 9% (9/100);
- Se a quantidade (em m³) de toras for **igual ou maior** que 1000 e **menor ou igual** que 2000, o desconto será de 16% (16/100);
- Se a quantidade (em m³) de toras for **maior** que 2000, não é aceito pedidos com essa quantidade de toras;
  
- ♦ Para o **adicional** de transporte rodoviário (1) é cobrado um valor **extra** de 1000 reais;
- ♦ Para o **adicional** de transporte ferroviário (2) é cobrado um valor **extra** de 2000 reais;
- ♦ Para o **adicional** de transporte hidroviário (3) é cobrado um valor **extra** de 2500 reais;

O valor final da conta é calculado da seguinte maneira:

$$\text{total} = ((\text{tipoMadeira} * \text{qtdToras}) * (1 - \text{desconto})) + \text{transporte}$$

Elabore um programa em Python que:

- Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).  
Por exemplo: `print("Bem-vindos a Madeireira do Lenhador Bruno Kostiuk")`  
**[EXIGÊNCIA DE CÓDIGO 1 de 7]**
- Deve-se implementar a função **escolha\_tipo()** que **não** recebe parâmetros e que:  
**[EXIGÊNCIA DE CÓDIGO 2 de 7]**
  - Pergunta o **tipo de madeira** desejado;
  - Retorna o VALOR do tipo de madeira** com base na escolha do usuário (use **return**);
  - Repete a pergunta do item **B.a** se digitar uma opção diferente de: PIN/PER/MOG/IPE/IMB;

- C. Deve-se implementar a função **qtd\_toras()** que **não** recebe parâmetros e que:  
[EXIGÊNCIA DE CÓDIGO 3 de 7];
- a. Pergunta a **quantidade de toras**;
  - b. **Retorna** (use **return**) a **quantidade de toras E o valor do desconto** (os dois valores) seguindo a regra do enunciado;
  - c. Repete a pergunta do item **C.a** se digitar um valor acima de 2000 ou valor não numérico (use try/except para não numérico)
- D. Deve-se implementar a função **transporte()** que **não** recebe parâmetros e que:  
[EXIGÊNCIA DE CÓDIGO 4 de 7];
- a. Pergunta pelo serviço **adicional de transporte**;
  - b. **Retorna** (use **return**) o **valor** de apenas uma das **opções de transporte**;
  - c. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/3;
- E. Deve-se implementar o total a pagar no código principal (**main**), ou seja, não pode estar dentro de função, conforme o enunciado [EXIGÊNCIA DE CÓDIGO 5 de 7];
- F. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
- G. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
- H. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
- I. Deve-se apresentar na saída de console um pedido no qual o usuário errou a opção de tipo de madeira [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
- J. Deve-se apresentar na saída de console um pedido no qual o usuário digitou um valor que ultrapasse a quantidade máxima de toras aceitas (2000) [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
- K. Deve-se apresentar na saída de console um pedido com opção de tipo de madeira, quantidade de toras e transporte válidos [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

## EXEMPLO DE SAÍDA DE CONSOLE:

```
➡ Bem vindo a Madeireira do Lenhador Bruno Kostiuk Nome completo

Entre com o Tipo de Madeira desejado
PIN - Tora de Pinho
PER - Tora de Peroba
MOG - Tora de Mogno
IPE - Tora de Ipê
IMB - Tora de Imbuia Errou o tipo de Madeira
>>TÁBUA
Escolha inválida, entre com o modelo novamente

Entre com o Tipo de Madeira desejado
PIN - Tora de Pinho
PER - Tora de Peroba
MOG - Tora de Mogno
IPE - Tora de Ipê
IMB - Tora de Imbuia
>>IPE Errou a quantidade de toras
Entre com a quantidade de toras (m³): 500000
Não aceitamos pedidos com essa quantidade de toras.
Por favor, entre com a quantidade novamente.

Entre com a quantidade de toras (m³): 500

Escolha o tipo de Transporte:
1 - Transporte Rodoviário - R$ 1000.00
2 - Transporte Ferroviário - R$ 2000.00
3 - Transporte Hidroviário - R$ 2500.00
>>3
Total: R$ 98095.50 Pedido com tipo de tora, quantidade de tora e transporte válidos
```

Figura 3.1: Exemplo de saída de console que o aluno deve fazer. Em que se pergunta pelo tipo de tora e se erra opção inicialmente, e que se passa a quantidade de toras acima do aceito. Na sequência, o usuário digitou um tipo de tora, quantidade de toras e transporte válidos.

### ***Apresentação de Código da Questão 3:***

```
print ( '|', '-' * 8, 'Bem-vindos a Madeireira do Lenhador Olavo Molina', '-' * 9, '|')
print ( '|', '-' * 42, '|' '\n')
```

#Função para escolher o tipo de madeira retornar o valor da madeira escolhida.

```
def escolha_tipo():
    while True:
        tipo_madeira = input('Escolha o tipo de madeira:"\n'
            'PIN - Tora de Pinho"\n'
            'PER - Tora de Peroba"\n'
            'MOG - Tora de Mogno"\n'
            'IPE - Tora de Ipê"\n'
            'IMB - Tora de Imbuia"\n' '>>').upper()
        if tipo_madeira == 'PIN':
            return 150.40
        elif tipo_madeira == 'PER':
            return 170.20
        elif tipo_madeira == 'MOG':
            return 190.90
        elif tipo_madeira == 'IPE':
            return 210.10
        elif tipo_madeira == 'IMB':
            return 220.70
        else:
            print('Tipo de madeira inválido. Tente novamente."\n')
```

# Função para obter a quantidade de toras e informar o desconto.

```
def qtd_toras():
    while True:
        try:
            qtd_toras = int(input("\nInforme a quantidade de toras (m3):"\n">>'))
            if qtd_toras < 0:
                print('A quantidade tem que ser maior que 0. Tente novamente."\n')
            elif qtd_toras > 2000:
                print('Quantidade de toras excedida. Limite máximo de 2000 toras. Tente novamente."\n')
            else:
                if qtd_toras < 100:
                    desconto = 0
                elif 100 <= qtd_toras < 500:
                    desconto = 4/100
                elif 500 <= qtd_toras < 1000:
                    desconto = 9/100
                elif 1000 <= qtd_toras <= 2000:
                    desconto = 16/100
                return qtd_toras, desconto
        except ValueError:
```

```
print('Valor inválido. Por favor, insira um número.\n')
```

```
# Função para escolher o tipo de transporte e informar o custo do transporte.
```

```
def transporte():
```

```
    while True:
```

```
        try:
```

```
            tipo_transporte = int(input('\n"Escolha o tipo de transporte:"\n'
```

```
            '1 - Rodoviário" -" R$ 1000"\n'
```

```
            '2 - Ferroviário"- " R$ 2000"\n'
```

```
            '3 - Hidroviário"- " R$ 2500"\n' '>>'))
```

```
            if tipo_transporte == 1:
```

```
                return 1000
```

```
            elif tipo_transporte == 2:
```

```
                return 2000
```

```
            elif tipo_transporte == 3:
```

```
                return 2500
```

```
            else:
```

```
                print('Tipo de transporte inválido. Tente novamente.\n')
```

```
        except ValueError:
```

```
            print('Valor inválido. Por favor, insira um número inteiro.\n')
```

```
# Cálculo do total a pagar usando (main)
```

```
if __name__ == "__main__":
```

```
    # Busca o valor do tipo de madeira escolhido.
```

```
    valor_madeira = escolha_tipo()
```

```
    # Busca a quantidade de toras e o desconto aplicável.
```

```
    qtd_toras, desconto = qtd_toras()
```

```
    # Busca o valor do transporte escolhido.
```

```
    tipo_transporte = transporte()
```

```
    # Calcula o valor total a pagar.
```

```
    total = (( valor_madeira * qtd_toras) * (1 - desconto)) + tipo_transporte
```

```
    # Exibindo o resultado final.
```

```
    print('\n"f"Valor total a pagar: R$ {total:.2f}')
```

### ***Apresentação de Saída do Console da Questão 3:***

```
| - - - - - Bem-vindos a Madeireira do Lenhador Olavo Molina - - - - - |  
| - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - |
```

Escolha o tipo de madeira:

PIN - Tora de Pinho

PER - Tora de Peroba

MOG - Tora de Mogno

IPE - Tora de Ipê

IMB - Tora de Imbuia

>>PIG

Tipo de madeira inválido. Tente novamente.

Escolha o tipo de madeira:

PIN - Tora de Pinho

PER - Tora de Peroba

MOG - Tora de Mogno

IPE - Tora de Ipê

IMB - Tora de Imbuia

>>PIN

Informe a quantidade de toras (m3):

>>3000

Quantidade de toras excedida. Limite máximo de 2000 toras. Tente novamente.

Informe a quantidade de toras (m3):

>>300

Escolha o tipo de transporte:

1 - Rodoviário - R\$ 1000

2 - Ferroviário- R\$ 2000

3 - Hidroviário- R\$ 2500

>>1

Valor total a pagar: R\$ 44315.20

#### QUESTÃO 4 de 4 - Conteúdo até aula 06

Enunciado: Você e sua equipe de programadores foram contratados por uma pequena empresa para desenvolver um software de gerenciamento de Contatos Comerciais. Este software deve ter o seguinte menu e opções:

- 1) Cadastrar Contato
- 2) Consultar Contato
  1. Consultar Todos
  2. Consultar por Id
  3. Consultar por Atividade
  4. Retornar ao menu
- 3) Remover Contato
- 4) Encerrar Programa

Elabore um programa em Python que:

- A. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).  
Por exemplo: **print("Bem vindos a lista de contatos do Bruno Kostiuk")** [EXIGÊNCIA DE CÓDIGO 1 de 8];
- B. Deve-se implementar uma lista com o nome de **lista\_contatos** e a variável **id\_global** com valor inicial igual ao número de seu RU [EXIGÊNCIA DE CÓDIGO 2 de 8];
- C. Deve-se implementar uma função chamada **cadastrar\_contato(id)** que recebe **apenas id** como parâmetro e que: [EXIGÊNCIA DE CÓDIGO 3 de 8];
  - a. Pergunta **nome**, **atividade**, **telefone** do contato;
  - b. Armazena o **id** (este é fornecido via parâmetro da função), **nome**, **atividade**, **telefone** dentro de um dicionário;
  - c. **Copiar** o dicionário para dentro da **lista\_contatos** (utilizar o **copy**);
- D. Deve-se implementar uma função chamada **consultar\_contatos()** que **não** recebe parâmetros e que: [EXIGÊNCIA DE CÓDIGO 4 de 8];
  - a. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
    - i. Se Consultar Todos, apresentar todos os contatos com todos os seus dados cadastrados;
    - ii. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o contato **específico** (apenas 1) com todos os seus dados cadastrados;
    - iii. Se Consultar por Atividade, solicitar ao usuário que informe a atividade, e apresentar o(s) contato(s) que exercem aquela atividade com todos os seus dados cadastrados;
    - iv. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
    - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **D.a**.
    - vi. Enquanto o usuário não escolher a opção 4, o menu consultar contatos deve se repetir.
- E. Deve-se implementar uma função chamada **remover\_contato()** em que: [EXIGÊNCIA DE CÓDIGO 5 de 8];

- a. Deve-se pergunta pelo **id** do contato a ser removido;
  - b. Remover o contato da **lista\_contatos**;
  - c. Se o id fornecido não for de um contato da lista, imprimir “**Id inválido**” e repetir a pergunta **E.a**.
- F. Deve-se implementar uma estrutura de menu no código principal (**main**), ou seja, **não pode estar dentro de função**, em que: **[EXIGÊNCIA DE CÓDIGO 6 de 8]**:
- a. Deve-se pergunta qual opção deseja (1. Cadastrar Contato / 2. Consultar Contato / 3. Remover Contato / 4. Encerrar Programa):
    - i. Se Cadastrar Contato, **incrementar** em um **id\_global** e **em seguida**, chamar a função **cadastrar\_contato (id\_global)**;
    - ii. Se Consultar Contato, chamar função **consultar\_contato ()**;
    - iii. Se Remover Contato, chamar função **remover\_contato ()**;
    - iv. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
    - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, imprimir “Opção inválida” e repetir a pergunta **F.a**.
    - vi. Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
- G. Deve-se implementar uma **lista de dicionários** (uma lista contendo dicionários dentro) **[EXIGÊNCIA DE CÓDIGO 7 de 8]**:
- H. Deve-se inserir comentários **relevantes** no código **[EXIGÊNCIA DE CÓDIGO 8 de 8]**:
- I. Deve-se apresentar na saída de console um cadastro do **seu contato** da seguinte forma: para **nome** informe seu **nome completo** (não usar apelidos ou abreviações), para **atividade** informar como **estudante**, e para **telefone** informe sua **RU**. **[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6]**;
  - J. Deve-se apresentar na saída de console um cadastro de **mais 2** contatos com mesmo tipo de atividade (por exemplo: marceneiro, padeiro, pintor, pedreiro) **[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 6]**;
  - K. Deve-se apresentar na saída de console uma consulta de todos os contatos **[EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6]**;
  - L. Deve-se apresentar na saída de console uma consulta por código (id) de um dos contatos **[EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6]**;
  - M. Deve-se apresentar na saída de console uma consulta por atividade em que **2** contatos exerçam a mesma atividade **[EXIGÊNCIA DE SAÍDA DE CONSOLE 5 de 6]**;
  - N. Deve-se apresentar na saída de console uma remoção de um dos contatos e em seguida de uma consulta de todos os contatos, provando que o contato foi removido **[EXIGÊNCIA DE SAÍDA DE CONSOLE 6 de 6]**;



## EXEMPLO DE SAÍDA DE CONSOLE:

```
➡ Bem vindo a Lista de Contatos do Bruno Kostiuk Nome completo
-----
----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Contato
2 - Consultar Contato(s)
3 - Remover Contato
4 - Sair
>>1
-----
----- MENU CADASTRAR CONTATO -----
Id do Contato: 4297914
Por favor entre com o nome do Contato: Bruno Kostiuk
Por favor entre com a Atividade do contato: Estudante
Por favor entre com o telefone do contato: 4297913
-----
```

**Figura 4.1: Exemplo de saída de console que o aluno deve fazer. Apresenta o print com seu nome completo e é realizado o cadastro do primeiro contato, note que o ID do contato não inicia em 1, pois ele deve iniciar com o seu RU (caso o RU informado não seja o seu, irá receber zero em toda questão). O primeiro contato deve ser cadastrado com SEU NOME COMPLETO, em Atividade informe Estudante e em Contato informe o SEU RU.**

```
➡ -----
----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Contato
2 - Consultar Contato(s)
3 - Remover Contato
4 - Sair
>>1
-----
----- MENU CADASTRAR CONTATO -----
Id do Contato: 4297915
Por favor entre com o nome do Contato: Tamy
Por favor entre com a Atividade do contato: Professor
Por favor entre com o telefone do contato: 99998888
-----
----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Contato
2 - Consultar Contato(s)
3 - Remover Contato
4 - Sair
>>1
-----
----- MENU CADASTRAR CONTATO -----
Id do Contato: 4297916
Por favor entre com o nome do Contato: Osmar
Por favor entre com a Atividade do contato: Professor
Por favor entre com o telefone do contato: 88889999
-----
```

**Figura 4.2: Exemplo de saída de console que o aluno deve fazer. São cadastrados mais dois contatos com mesmo tipo de Atividade.**

```
-----  
----- MENU CONSULTAR CONTATOS -----  
Escolha a opção desejada:  
1 - Consultar Todos os Contatos  
2 - Consultar Contato por id  
3 - Consultar Contato(s) por Atividade  
4 - Retornar  
>>1  
-----  
id: 4297914  
nome: Bruno KostiuK  
atividade: Estudante  
telefone: 4297913  
  
id: 4297915  
nome: Tamy  
atividade: Professor  
telefone: 99998888  
  
id: 4297916  
nome: Osmar  
atividade: Professor  
telefone: 88889999  
-----
```

Figura 4.3: Exemplo de saída de console que o aluno deve fazer. Em que se consulta Todos os contatos cadastrados.

```
⇒ Escolha a opção desejada:
1 - Consultar Todos os Contatos
2 - Consultar Contato por id
3 - Consultar Contato(s) por Atividade
4 - Retornar
>>2
Digite o id do contato: 4297914
-----
id: 4297914
nome: Bruno KostiuK
atividade: Estudante
telefone: 4297913

-----

----- MENU CONSULTAR CONTATOS -----
Escolha a opção desejada:
1 - Consultar Todos os Contatos
2 - Consultar Contato por id
3 - Consultar Contato(s) por Atividade
4 - Retornar
>>3
Digite a Atividade do(s) Contato(s): Professor
-----
id: 4297915
nome: Tamy
atividade: Professor
telefone: 99998888

id: 4297916
nome: Osmar
atividade: Professor
telefone: 88889999

-----
```

**Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o contato com id número 4297914 e consulta pelo nome da Atividade (Professor).**

```
-----  
----- MENU PRINCIPAL -----  
Escolha a opção desejada:  
1 - Cadastrar Contato  
2 - Consultar Contato(s)  
3 - Remover Contato  
4 - Sair  
Remove um contato  
>>3  
-----  
----- MENU REMOVER CONTATO -----  
Digite o id do contato a ser removido: 4297914  
Contato removido com sucesso!  
-----  
----- MENU PRINCIPAL -----  
Escolha a opção desejada:  
1 - Cadastrar Contato  
2 - Consultar Contato(s)  
3 - Remover Contato  
4 - Sair  
>>2  
-----  
----- MENU CONSULTAR CONTATOS -----  
Escolha a opção desejada:  
1 - Consultar Todos os Contatos  
2 - Consultar Contato por id  
3 - Consultar Contato(s) por Atividade  
4 - Retornar  
>>1  
-----  
id: 4297915  
nome: Tamy  
atividade: Professor  
telefone: 99998888  
Realiza o consultar Todos mostrando  
que o contato foi removido  
  
id: 4297916  
nome: Osmar  
atividade: Professor  
telefone: 88889999  
-----
```

**Figura 4.5: Exemplo de saída de console que o aluno deve fazer. Em que se remove o contato de Id número 4297914 e depois se faz uma consulta de todos os contatos.**

*Apresentação de **Código da Questão 4:***

```
print ( '|', '-' * 8, 'Bem-vindos a lista de contatos do Olavo Molina', '-' * 8, '|')
```

```

print ( '|', '-' * 40, '|' '\n')

# Lista para armazenar contatos e id_global
lista_contatos = []
id_global = 4994799

# Função para cadastrar contato
def cadastrar_contato(id):
    nome = input('\nInforme o nome do contato: ')
    atividade = input('Informe a atividade do contato: ')
    telefone = input('Informe o telefone do contato: ')

    # Cria o dicionário com as informações do contato
    contato = {'id': id, 'nome': nome, 'atividade': atividade, 'telefone': telefone}

    # Adiciona o dicionário na lista de contatos
    lista_contatos.append(contato.copy())

# Função para consultar contatos por opções.
def consultar_contatos():
    while True:
        print('\nQual opção você deseja:')
        print('1. Consultar todos')
        print('2. Consultar por Id')
        print('3. Consultar por atividade')
        print('4. Retornar ao menu' '\n')

        opcao = input('\nDigite o número opção desejada: ')
        # Consultar todos os contatos
        if opcao == '1':
            if not lista_contatos:
                print('Nenhum contato cadastrado.')
            else:
                for contato in lista_contatos:
                    print(f"\nID: {contato['id']}, Nome: {contato['nome']}, Atividade: {contato['atividade']},
Telefone: {contato['telefone']}")
        # Consultar contato por ID
        elif opcao == '2':
            id_consulta = int(input('Informe o ID do contato: '))
            # Pesquisa se o ID está na lista
            contato_encontrado = None
            for contato in lista_contatos:
                if contato['id'] == id_consulta:
                    contato_encontrado = contato
                    break
            # Mostra os dados do ID encontrado ou Id inválido.
            if contato_encontrado:

```

```

        print(f"ID: {contato_encontrado['id']}, Nome: {contato_encontrado['nome']}, Atividade: {contato_encontrado['atividade']}, Telefone: {contato_encontrado['telefone']}")
    else:
        print('Id inválido. Tente novamente.')
# Consultar contato por Atividade
elif opcao == '3':
    atividade_consulta = input('Informe a atividade: ')
    encontrou_contato = False
    for contato in lista_contatos:
        if contato['atividade'].lower() == atividade_consulta.lower():
            print(f"ID: {contato['id']}, Nome: {contato['nome']}, Atividade: {contato['atividade']}, Telefone: {contato['telefone']}")
            encontrou_contato = True
    if not encontrou_contato:
        print('Nenhum contato encontrado para esta atividade.')
# Retornar ao menu principal
elif opcao == '4':
    return
# Consulta com opção inválida
else:
    print('Opção inválida.')

# Função para remover contato
def remover_contato():
    while True:
        id_remove = int(input("\nInforme o ID do contato a ser removido: "))
        # Pesquisa se o ID está na lista
        contato_encontrado = None
        for contato in lista_contatos:
            if contato['id'] == id_remove:
                contato_encontrado = contato
                break
        # Remove o ID encontrado da lista de contatos
        if contato_encontrado:
            lista_contatos.remove(contato_encontrado)
            print(f"Contato com ID {id_remove} removido com sucesso.")
            break
        # Consulta com ID inválido.
        else:
            print('Id inválido.')

# Estrutura do menu de entrada
while True:
    print( "\n|", '-' * 9, 'Menu Principal', '-' * 9, '|', )
    print( '|', '-' * 34, '|', '\n')
    print('1. Cadastrar Contato')
    print('2. Consultar Contato')

```

```

print('3. Remover Contato')
print('4. Encerrar Programa')

opcao = input("\nDigite o número opção desejada: ")
# Chama função para cadastrar contato e incrementa o id_global
if opcao == '1':
    id_global += 1
    cadastrar_contato(id_global)
# Chama função para cadastrar contatos.
elif opcao == '2':
    consultar_contatos()
# Chama função para remover contatos.
elif opcao == '3':
    remover_contato()
# Encerra o programa.
elif opcao == '4':
    print('Encerrando o programa...')
    break
else:
    print('Opção inválida.')

```

#### **Apresentação de Saída do Console da Questão 4:**

##### **Saída de Console 1:**

```

... | - - - - - Bem-vindos a lista de contatos do Olavo Molina - - - - - |
    | - - - - - - - - - - - - - - - - - - - - - - - - - - - - - |

    | ----- Menu Principal ----- |
    | ----- |

1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 1

    | ----- Menu Cadastrar Contatos ----- |
    | ----- |

ID do Contato: 4994800
Informe o nome do contato: Olavo Guimaraes Molina
Informe a atividade do contato: Estudante
Informe o telefone do contato: 4994799

```



## Saída de Console 2:

\*\*\*

```
| ----- Menu Principal ----- |  
| ----- |
```

1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 1

```
| ----- Menu Cadastrar Contatos ----- |  
| ----- |
```

ID do Contato: 4994801  
Informe o nome do contato: Joao Antonio  
Informe a atividade do contato: Professor  
Informe o telefone do contato: 65981144556

```
| ----- Menu Principal ----- |  
| ----- |
```

1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 1

```
| ----- Menu Cadastrar Contatos ----- |  
| ----- |
```


ID do Contato: 4994802  
Informe o nome do contato: Carlos Augusto  
Informe a atividade do contato: Professor  
Informe o telefone do contato: 65981144543

---



### Saída de Console 3:

```

 | ----- Menu Principal ----- |
... | -----
1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 2

| ----- Menu Consultar Contatos ----- |
| -----

Qual opção você deseja:
1. Consultar todos
2. Consultar por Id
3. Consultar por atividade
4. Retornar ao menu

Digite o número opção desejada: 1

ID: 4994800, Nome: Olavo Guimaraes Molina, Atividade: Estudante, Telefone: 4994799
ID: 4994801, Nome: Joao Antonio, Atividade: Professor, Telefone: 65981144556
ID: 4994802, Nome: Carlos Augusto, Atividade: Professor, Telefone: 65981144543
```

### Saída de Console 4 e 5:



...

```
| ----- Menu Consultar Contatos ----- |  
| ----- |
```

Qual opção você deseja:

1. Consultar todos
2. Consultar por Id
3. Consultar por atividade
4. Retornar ao menu

Digite o número opção desejada: 2

Informe o ID do contato: 4994800

ID: 4994800, Nome: Olavo Guimaraes Molina, Atividade: Estudante, Telefone: 4994799

```
| ----- Menu Consultar Contatos ----- |  
| ----- |
```

Qual opção você deseja:

1. Consultar todos
2. Consultar por Id
3. Consultar por atividade
4. Retornar ao menu

Digite o número opção desejada: 3

Informe a atividade: Professor

ID: 4994801, Nome: Joao Antonio, Atividade: Professor, Telefone: 65981144556

ID: 4994802, Nome: Carlos Augusto, Atividade: Professor, Telefone: 65981144543

### Saída de Console 6:

```
| ----- Menu Principal ----- |  
| ----- |
```

1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 3

```
| ----- Menu Remover Contatos ----- |  
| ----- |
```

Informe o ID do contato a ser removido: 4994800  
Contato com ID 4994800 removido com sucesso.

```
| ----- Menu Principal ----- |  
| ----- |
```

1. Cadastrar Contato
2. Consultar Contato
3. Remover Contato
4. Encerrar Programa

Digite o número opção desejada: 2

```
| ----- Menu Consultar Contatos ----- |  
| ----- |
```

Qual opção você deseja:

1. Consultar todos
2. Consultar por Id
3. Consultar por atividade
4. Retornar ao menu

Digite o número opção desejada: 1

ID: 4994801, Nome: Joao Antonio, Atividade: Professor, Telefone: 65981144556

ID: 4994802, Nome: Carlos Augusto, Atividade: Professor, Telefone: 65981144543