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:

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

Elabore um programa em Python que:

1. 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];

1. Deve-se implementar o input do **valorBase** do plano e da **idade** do cliente [EXIGÊNCIA DE CÓDIGO 2 de 6];
2. 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];
3. Deve-se implementar o **valorMensal** [EXIGÊNCIA DE CÓDIGO 4 de 6];
4. Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
5. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
6. Deve-se apresentar na saída de console uma mensagem com seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];
7. 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:

A imagem apresenta o seguinte texto como 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 Alessandro Gomes Soares')

#Solicita ao usuário que insira o valor base do seu plano.

valorBase = float(input('Digite o valor base do plano: R$'))

#Solicita ao usuário que insira a sua idade.

idade = int(input('Digite a idade do cliente: '))

#Parâmetros para o cálculo da porcentagem, baseado na idade informada.

if (idade >= 0 and idade < 19):

    porcentagem = 1 #100/100

elif (idade >= 19 and idade < 29):

    porcentagem = 1.5 #150/100

elif (idade >= 29 and idade < 39):

    porcentagem = 2.25 #225/100

elif (idade >= 39 and idade < 49):

    porcentagem = 2.40 #240/100

elif (idade >= 49 and idade < 59):

    porcentagem = 3.5 #350/100

else:

    porcentagem = 6.0 #600/100

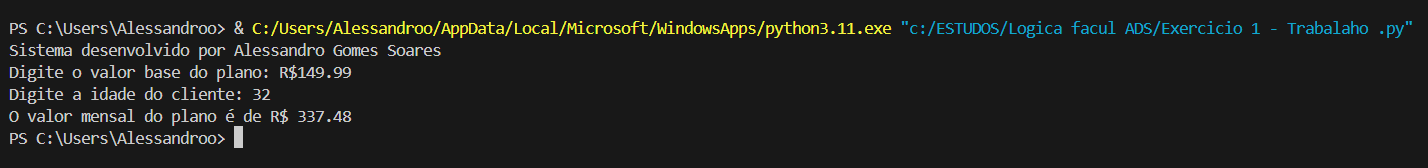
#Realiza o cálculo do valor a ser pago, de acordo com o valor base e a porcentagem relacionada a idade.

valorMensal = valorBase \* porcentagem

#Exibi o valor a ser pago em reais.

print(f'O valor mensal do plano é de R$ {valorMensal:.2f}')

Apresentação de **Saída do Console da Questão 1**:



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:

1. 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];

1. 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];
2. 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];
3. 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];
4. Deve-se implementar um **acumulador** para somar os valores dos pedidos (valor total do pedido) [EXIGÊNCIA DE CÓDIGO 5 de 8];
5. 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];
6. Deve-se implementar as estruturas de **while, break, continue (todas elas)** [EXIGÊNCIA DE CÓDIGO 7 de 8];
7. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
8. 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];
9. Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
10. Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
11. 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:

A imagem apresenta o seguinte texto como 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

**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**:

#Menu, para o cliente visualizar os tamanhos e valores.

print('-'\*10,'Bem-vindo Pizzaria do Alessandro Gomes Soares','-'\*10)

print('-'\*29,'Cardápio','-'\*28)

print('-'\*67)

print('-'\*3,'|  Tamanho   |   Pizza Salgada (PS)  |   Pizza Doce (PD)  |','-'\*3)

print('-'\*3,'|     P      |       R$ 30.00        |      R$ 34.00      |','-'\*3)

print('-'\*3,'|     M      |       R$ 45.00        |      R$ 48.00      |','-'\*3)

print('-'\*3,'|     G      |       R$ 60.00        |      R$ 66.00      |','-'\*3)

print('-'\*67)

#Variável para guardar os valores dos pedidos.

total = 0

#início do laço de repetição do menu

while True:

    #entrada do sabor

    sabor = str(input('Qual sabor desejado (PD)/(PS): ')).upper()

    #validação dos sabores

    if sabor != 'PD' and sabor != 'PS':

        print('Sabor inválido. tente novamente.')

        continue

    #entrada do tamanho

    tamanho = str(input('Qual tamanho desejado (P/M/G): ')).upper()

    #validação dos tamanhos

    if tamanho != 'P' and tamanho != 'M' and tamanho != 'G':

        print('tamanho inválido. tente novamente.')

        continue

    #verifica qual valor a ser cobrado de acordo com o tamanho da pizza doce

    if sabor == 'PD':

        if tamanho == 'P':

            total += 34.00

            print(f'Pizza Doce, tamanho pequena adicionada: R$ 34.00')

        elif tamanho == 'M':

            total += 48.00

            print(f'Pizza Doce, tamanho média adicionada:R$ 48.00')

        else:

            total += 66.00

            print(f'Pizza Doce, tamanho grande adicionada:R$ 66.00')

    #verifica qual valor a ser cobrado de acordo com o tamanho da pizza salgada

    else:

        if tamanho == 'P':

            total += 30.00

            print(f'Pizza Salgada, tamanho pequena adicionada:R$ 30.00')

        elif tamanho == 'M':

            total += 45.00

            print(f'Pizza Salgada, tamanho média adicionada:R$ 45.00')

        else:

            total += 60.00

            print(f'Pizza Salgada, tamanho grande adicionada:R$ 60.00')

    #verifica se o cliente quer adicionar mais itens

    resposta = str(input('Deseja continuar com o pedido? (S/N): ')).upper()

    #se resposta for diferente de S o pedido é finalizado

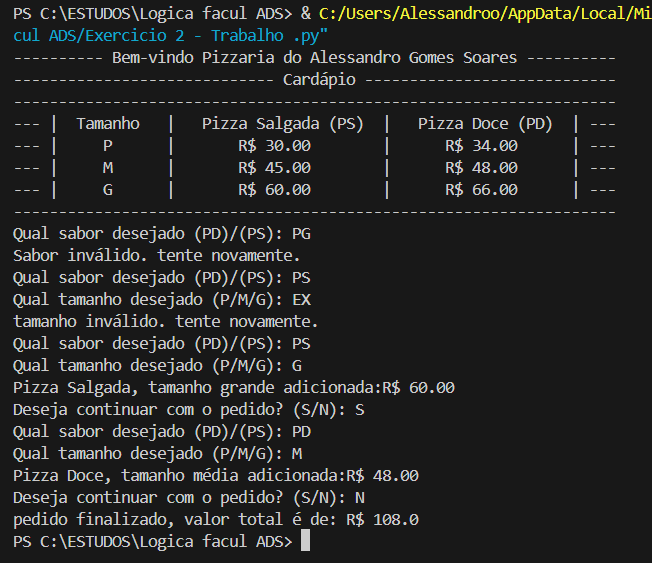
    if resposta != 'S':

        break

#amostra o valor do pedido

print(f'pedido finalizado, valor total é de: R$ {total}')

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



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 arvore 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:

total = ((**tipoMadeira \* qtdToras)\*(1-desconto)) + transporte**

Elabore um programa em Python que:

1. 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];

1. Deve-se implementar a função **escolha\_tipo()** que **não** recebe parâmetros e que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Pergunta o **tipo de madeira** desejado;
   2. **Retorna** **o VALOR** **do tipo de madeira** com base na escolha do usuário (use **return**);
   3. Repete a pergunta do item **B.a** se digitar uma opção diferente de: PIN/PER/MOG/IPE/IMB;
2. Deve-se implementar a função **qtd\_toras()** que **não** recebe parâmetros e que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Pergunta a **quantidade de toras**;
   2. **Retorna** (use **return**) a **quantidade de toras E o valor do desconto** (os dois valores) seguindo a regra do enunciado;
   3. Repete a pergunta do item **C.a** se digitar um valor acima de 2000 ou valor não numérico (use try/except para não numérico)
3. Deve-se implementar a função **transporte()** que **não** recebe parâmetros e que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Pergunta pelo serviço **adicional de transporte**;
   2. **Retorna** (use **return**) o **valor** de apenas uma das **opções** de **transporte**;
   3. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/3;
4. 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];
5. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
6. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
7. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
8. 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];
9. 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];
10. 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:

A imagem apresenta o seguinte texto como saída de console:

Bem vindo a Madeireira do Lenhador Bruno Kostiuk

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
>>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
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

**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('Bem-vindos a madereira do Alessandro Gomes Soares')

#função para escolher o tipo de madeira e retonar o valor

def escolha\_tipo():

    while True:

        #mostra o tipo de madeira e seus valores

        print('entre com o tipo de madeira desejado ? ')

        print('PIN - Tora de pinho m³  - R$ 150.40')

        print('PER - Tora de peroba m³ - R$ 170.20')

        print('MOG - Tora de mogno m³  - R$ 190.90')

        print('IPE - Tora de ipê m³    - R$ 210.10')

        print('IMB - Tora de imbuia m³ - R$ 220.70')

        #armazena o tipo de madeira escolhido em maiúsculo

        tipo = input('>> ').upper()

        #verifica se a madeira escolhida é válida

        if tipo != 'PIN' and  tipo != 'PER' and tipo != 'MOG' and tipo != 'IPE' and tipo != 'IMB':

            print('escolha inválida, entre com o modelo novamente')

            continue

        #retorna o valor de acordo com a madeira escolhida

        elif tipo == 'PIN':

            return 150.40

        elif tipo == 'PER':

            return 170.20

        elif tipo == 'MOG':

            return 190.90

        elif tipo == 'IPE':

            return 210.10

        else:

            return 220.70

#função que determina o valor de desconto a ser aplicado

def qtd\_toras():

    while True:

     #vai tentar executar os if e elses

     try:

        #recebe a quantidade de toras e o valor de desconto

        qtd = int(input('infome a quatidade de toras (m³) ? '))

        if qtd >= 0 and qtd < 100:

            return qtd, 0.00

        elif qtd >= 100 and qtd < 500:

            return qtd, 0.04

        elif qtd >= 500 and qtd < 1000:

            return qtd, 0.09

        elif qtd >= 1000 and qtd <= 2000:

            return qtd, 0.16

        else:

            print('Não é aceito pedidos com essa quantidade de toras.\nEntre com a quantidade de toras novmente.')

    #caso não consiga executar os if e elses exibira o print

     except:

         print('entre com a quantidade em números inteiros')

#função para escolher o tipo de transporte e retornar o valor correspondente

def tipo\_transporte():

    while True:

        #mostra as opções de transporte e seu custo

        print('Qual tipo de transporte desejado ?: ')

        print('1 - transporte rodoviário  - R$ 1000.00')

        print('2 - transporte ferroviário - R$ 2000.00')

        print('3 - transporte hidroviário - R$ 2500.00')

        #recebe o valor do transporte

        trans = int(input('>> '))

        #valida se o valor do transporte foi inserido corretamente

        if trans != 1 and trans != 2 and trans != 3:

            print('forma de transporte inválida, tente novamente')

        elif trans == 1:

            return 1000

        elif trans == 2:

            return 2000

        else:

            return 2500

#chama as funções para obter o preço da madeira, quantidade com desconto, e tipo de transporte

madeira = escolha\_tipo()

quantidade, desconto = qtd\_toras()

transporte = tipo\_transporte()

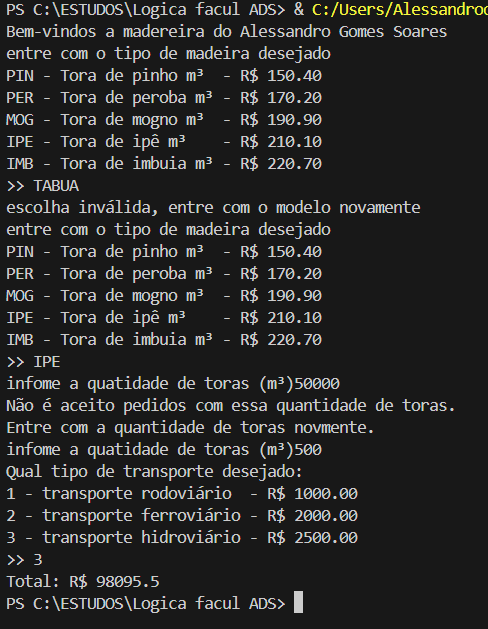
#calcula o valor total com base no preço da madeira, quantidade, desconto e custo do transporte

total = ((madeira \* quantidade) \* (1-desconto)) + transporte

#mostra o valor total do pedido

print(f'Total: R$ {total} ')

Apresentação de Saída do Console da Questão 3:



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:

1. 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];

1. 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];
2. Deve-se implementar uma função chamada **cadastrar\_contato(id)** que recebe **apenas id** como parâmetro e que**:** [EXIGÊNCIA DE CÓDIGO 3 de 8];
   1. Pergunta **nome**, **atividade**, **telefone** do contato;
   2. Armazena o **id** (este é fornecido via parâmetro da função), **nome**, **atividade**, **telefone** dentro de um dicionário;
   3. **Copiar** o dicionário para dentro da **lista\_contatos** (utilizar o **copy**);
3. Deve-se implementar uma função chamada **consultar\_contatos()** que **não** recebe parâmetros e que**:** [EXIGÊNCIA DE CÓDIGO 4 de 8];
   1. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
      1. Se Consultar Todos, apresentar todos os contatos com todos os seus dados cadastrados;
      2. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o contato **específico** (apenas 1) com todos os seus dados cadastrados;
      3. 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;
      4. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **D.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu consultar contatos deve se repetir.
4. Deve-se implementar uma função chamada **remover\_contato()** em que**:** [EXIGÊNCIA DE CÓDIGO 5 de 8];
   1. Deve-se pergunta pelo **id** do contato a ser removido;
   2. Remover o contato da **lista\_contatos**;
   3. Se o id fornecido não for de um contato da lista, printar “**Id inválido**” e repetir a pergunta **E.a**.
5. 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];
   1. Deve-se pergunta qual opção deseja (1. Cadastrar Contato / 2. Consultar Contato / 3. Remover Contato / 4. Encerrar Programa):
      1. Se Cadastrar Contato, **incrementar** em um **id\_ global** e **em seguida**, chamar a função **cadastrar\_contato (id\_ global)**;
      2. Se Consultar Contato, chamar função **consultar\_contato ()**;
      3. Se Remover Contato, chamar função **remover\_ contato ()**;
      4. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **F.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
6. Deve-se implementar uma **lista de dicionários** (uma lista contento dicionários dentro)[EXIGÊNCIA DE CÓDIGO 7 de 8];
7. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
8. 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];
9. 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];
10. Deve-se apresentar na saída de console uma consulta de todos os contatos [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6];
11. Deve-se apresentar na saída de console uma consulta por código (id) de um dos contados [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6];
12. 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];
13. 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:

A imagem apresenta o seguinte texto como saída de console:

Bem vindo a Lista de Contatos do Bruno Kostiuk
--------------------------------------------------
--------------- 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.**

A imagem apresenta o seguinte texto como saída de console:

--------------------------------------------------
--------------- 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.**

A imagem apresenta o seguinte texto como saída de console:

--------------------------------------------------
---------- 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.**

A imagem apresenta o seguinte texto como saída de console:

--------------------------------------------------
---------- 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
>>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).

A imagem apresenta o seguinte texto como saída de console:

--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Contato
2 - Consultar Contato(s)
3 - Remover Contato
4 - Sair
>>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

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('Bem vindo a lista de contatos de Alessandro Gomes Soares')

#Lista para armazenamento dos contatos

lista\_contatos = []

#variavel que recebe id dos contato

id\_global = 4976113

#fuçao para cadastrar novo contato

def cadastrar\_contato(id):

    print('-' \*42)

    print('-' \*13,' CADASTRO ', '-' \*13)

    print(f'Id do contato: {id}')

    #solicita as informaçoes do contato

    nome = input('Informe o nome do contato: ')

    telefone = input('Informe o telefone do contato: ')

    atividade = input ('Informe a atividade do contato: ')

    #dicionario que recebe as informaçoes do contato

    dic = {}

    dic ['id'] = id

    dic ['nome'] = nome

    dic ['atividade'] = atividade

    dic ['telefone'] = telefone

    print(dic)

    #adiciona uma copia do dicionario na lista de contatos

    lista\_contatos.append(dic.copy())

#funçao para consultar contatos

def consultar\_contatos():

    while True:

        print('-' \*42)

        print('-' \*13,' CONSULTA ', '-' \*13)

        print('Escolha a opção desejada:')

        #mostra as opçoes de consulta

        print('1 - Consultar todos')

        print('2 - Consultar por Id')

        print('3 - Consultar por atividade')

        print('4 - Retornar ao menu')

        opc = input('>> ')

        #opção de retornar ao menu principal

        if opc == '4':

            return

        #consulta todos os contatos cadastrados

        elif opc == '1':

            print('Consultar todos')

            for contato in lista\_contatos:

                print(contato)

        #consulta por ID específico

        elif opc =='2':

            print('Consultar por id')

            id = int(input('informe um Id: '))

            for contato in lista\_contatos:

                if id == contato['id']:

                    print(contato)

        #consulta por atividade

        elif opc =='3':

            print('Consultar por atividade')

            atividade = input('informe uma atividade: ')

            for contato in lista\_contatos:

                if atividade == contato['atividade']:

                    print(contato)

         #caso o usuário escolha uma opção inválida

        else:

            print('Opção inválida. tente novamente')

#função para remover um contato

def remove\_contatos():

    print('-' \* 42)

    print('-' \* 13, 'REMOVE CONTATO', '-' \* 13)

    print('Remover contato')

    #solicita id do contato a ser removido

    id\_remover = int(input('Informe a Id do contato que deseja remover: '))

    contato\_encontrado = False

    #percorre a lista para encontrar e remover o contato

    for contato in lista\_contatos:

        if id\_remover == contato['id']:

            lista\_contatos.remove(contato)

            contato\_encontrado = True

            print(f"Contato com Id {id\_remover} removido com sucesso.")

            break

    #se nao encontrar o contato exibe este print

    if not contato\_encontrado:

        print('Id inválido. Contato não encontrado.')

#programa principal

while True:

    print('-' \*42)

    print('-' \*13,'MENU PRINCIPAL', '-' \*13)

    #opçoes do menu

    print('1 - Cadastrar contato')

    print('2 - Consultar contato')

    print('3 - Remover contato')

    print('4 - Encerrar programa')

    escolha = input('>> ')

     #opção de cadastrar um novo contato

    if escolha == '1':

        id\_global += 1

        cadastrar\_contato(id\_global)

    #opção de consultar contatos

    elif escolha == '2':

        consultar\_contatos()

    #opção de remover um contato

    elif escolha == '3':

        remove\_contatos()

    #opção de encerrar o programa

    elif escolha == '4':

        break

    #caso digite uma opçao nao presente no menu

    else:

        print('Opção inválida')

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



