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**:

# Sistema desenvolvido por Julia fonseca

print("Sistema desenvolvido por Julia Fonseca")  # EXIGÊNCIA DE CÓDIGO 1 de 6

# Função principal para calcular o valor mensal do plano de saúde

def calcular\_valor\_mensal():

    # Captura o valor base do plano e a idade do cliente

    valor\_base = float(input("Informe o valor base do plano: "))  # EXIGÊNCIA DE CÓDIGO 2 de 6

    idade = int(input("Informe a idade do cliente: "))  # EXIGÊNCIA DE CÓDIGO 2 de 6

    # Aplicação das regras para determinar a porcentagem do plano de acordo com a idade [EXIGÊNCIA DE CÓDIGO 3 de 6]

    if idade >= 0 and idade < 19:

        porcentagem = 100 / 100

    elif idade >= 19 and idade < 29:

        porcentagem = 150 / 100

    elif idade >= 29 and idade < 39:

        porcentagem = 225 / 100

    elif idade >= 39 and idade < 49:

        porcentagem = 240 / 100

    elif idade >= 49 and idade < 59:

        porcentagem = 350 / 100

    elif idade >= 59:

        porcentagem = 600 / 100

    else:

        print("Idade inválida.")

        return

    # Cálculo do valor mensal do plano com base no valor base e na porcentagem [EXIGÊNCIA DE CÓDIGO 4 de 6]

    valor\_mensal = valor\_base \* porcentagem

    # Exibe o valor mensal do plano

    print(f"O valor mensal do plano de saúde para a idade {idade} é: R$ {valor\_mensal:.2f}")

# Chamada da função principal

calcular\_valor\_mensal()

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

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

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**:

# Bem-vindo à Pizzaria de Julia fonseca

print("Bem-vindos a Pizzaria de Julia Fonseca")

# Menu de opções

def mostrar\_menu():

    print("\n\*\*\* MENU \*\*\*")

    print("Sabores disponíveis:")

    print("PS - Pizza Salgada")

    print("PD - Pizza Doce")

    print("\nTamanhos disponíveis:")

    print("P - Pequena")

    print("M - Média")

    print("G - Grande")

    print("\nPreços:")

    print("Tamanho P - PS: R$30, PD: R$34")

    print("Tamanho M - PS: R$45, PD: R$48")

    print("Tamanho G - PS: R$60, PD: R$66")

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# Função para capturar o pedido

def capturar\_pedido():

    # Lista de sabores e tamanhos válidos

    sabores\_validos = ["PS", "PD"]

    tamanhos\_validos = ["P", "M", "G"]

    # Verificando sabor

    while True:

        sabor = input("Escolha o sabor (PS para Salgada, PD para Doce): ").upper()

        if sabor not in sabores\_validos:

            print("Sabor inválido. Tente novamente.")

            continue

        break

    # Verificando tamanho

    while True:

        tamanho = input("Escolha o tamanho (P para Pequena, M para Média, G para Grande): ").upper()

        if tamanho not in tamanhos\_validos:

            print("Tamanho inválido. Tente novamente.")

            continue

        break

    # Retornando o pedido escolhido

    return sabor, tamanho

# Função para calcular o preço com base no sabor e tamanho

def calcular\_preco(sabor, tamanho):

    # Modelo aninhado de if/else para calcular o preço

    if sabor == "PS":

        if tamanho == "P":

            return 30

        elif tamanho == "M":

            return 45

        elif tamanho == "G":

            return 60

    elif sabor == "PD":

        if tamanho == "P":

            return 34

        elif tamanho == "M":

            return 48

        elif tamanho == "G":

            return 66

# Função principal para executar o pedido

def fazer\_pedido():

    total = 0  # Acumulador para o valor total do pedido

    while True:

        # Exibir menu ao cliente

        mostrar\_menu()

        # Capturar o pedido

        sabor, tamanho = capturar\_pedido()

        # Calcular o preço do pedido

        preco = calcular\_preco(sabor, tamanho)

        total += preco

        print(f"Você pediu uma pizza {sabor} de tamanho {tamanho}. Valor: R${preco}")

        # Perguntar se deseja fazer mais pedidos

        mais\_pedido = input("Deseja pedir mais alguma coisa? (S/N): ").upper()

        if mais\_pedido == "N":

            print(f"Total do pedido: R${total}")

            break

# Executando o programa

fazer\_pedido()

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

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

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:

# Bem-vindos à Madeireira da Lenhadora Julia Fonseca

print("Bem-vindos à Madeireira da Lenhadora Julia Fonseca")  # EXIGÊNCIA DE CÓDIGO 1 de 7

# Função para escolher o tipo de madeira

def escolha\_tipo():

    while True:

        tipo = input("Escolha o tipo de madeira (PIN/PER/MOG/IPE/IMB): ").upper()  # Pergunta o tipo de madeira [EXIGÊNCIA DE CÓDIGO 2a de 7]

        if tipo == "PIN":

            return 150.40  # Retorna o valor do m³ de acordo com o tipo escolhido [EXIGÊNCIA DE CÓDIGO 2b de 7]

        elif tipo == "PER":

            return 170.20

        elif tipo == "MOG":

            return 190.90

        elif tipo == "IPE":

            return 210.10

        elif tipo == "IMB":

            return 220.70

        else:

            print("Opção inválida. Tente novamente.")  # Mensagem de erro [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4]

# Função para escolher a quantidade de toras

def qtd\_toras():

    while True:

        try:

            qtd = int(input("Informe a quantidade de toras em m³: "))  # Pergunta a quantidade de toras [EXIGÊNCIA DE CÓDIGO 3a de 7]

            if qtd > 2000:

                print("Quantidade inválida. Não aceitamos pedidos com mais de 2000 m³.")  # Mensagem de erro [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4]

                continue

            elif qtd >= 1000:

                desconto = 16 / 100

            elif qtd >= 500:

                desconto = 9 / 100

            elif qtd >= 100:

                desconto = 4 / 100

            else:

                desconto = 0 / 100

            return qtd, desconto  # Retorna a quantidade e o desconto [EXIGÊNCIA DE CÓDIGO 3b de 7]

        except ValueError:

            print("Valor inválido. Por favor, insira um número.")  # Tratamento de erro para entrada não numérica [EXIGÊNCIA DE CÓDIGO 6 de 7]

# Função para escolher o tipo de transporte

def transporte():

    while True:

        opcao\_transporte = input("Escolha o tipo de transporte (1 - Rodoviário / 2 - Ferroviário / 3 - Hidroviário): ")  # Pergunta o transporte [EXIGÊNCIA DE CÓDIGO 4a de 7]

        if opcao\_transporte == "1":

            return 1000  # Transporte rodoviário

        elif opcao\_transporte == "2":

            return 2000  # Transporte ferroviário

        elif opcao\_transporte == "3":

            return 2500  # Transporte hidroviário

        else:

            print("Opção inválida. Tente novamente.")  # Mensagem de erro para transporte inválido

# Cálculo do total

tipo\_madeira\_valor = escolha\_tipo()  # Chama função para escolher o tipo de madeira

qtd, desconto = qtd\_toras()  # Chama função para escolher a quantidade e calcular o desconto

valor\_transporte = transporte()  # Chama função para escolher o transporte

# Calcula o total a pagar [EXIGÊNCIA DE CÓDIGO 5 de 7]

total = ((tipo\_madeira\_valor \* qtd) \* (1 - desconto)) + valor\_transporte

# Exibe o valor total

print(f"O valor total a pagar é: R$ {total:.2f}")  # Exibe o valor total [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4]

# Fim do programa

Apresentação de Saída do Console da Questão 3:

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

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**:

*# Bem-vindos à lista de contatos de Julia Fonseca*

*print("Bem-vindos à lista de contatos de Julia Fonseca")  # EXIGÊNCIA DE CÓDIGO 1 de 8*

*# Lista para armazenar os contatos*

*lista\_contatos = []  # EXIGÊNCIA DE CÓDIGO 7 de 8*

*# Variável global para gerar os IDs*

*id\_global = 12345678  # Coloque seu RU aqui [EXIGÊNCIA DE CÓDIGO 2 de 8]*

*# Função para cadastrar contatos*

*def cadastrar\_contato(id\_):*

*nome = input("Informe o nome do contato: ")  # Pergunta o nome do contato [EXIGÊNCIA DE CÓDIGO 3a de 8]*

*atividade = input("Informe a atividade: ")  # Pergunta a atividade*

*telefone = input("Informe o telefone: ")  # Pergunta o telefone*

*contato = {'id': id\_, 'nome': nome, 'atividade': atividade, 'telefone': telefone}  # Armazena em um dicionário [EXIGÊNCIA DE CÓDIGO 3b de 8]*

*lista\_contatos.append(contato.copy())  # Adiciona à lista de contatos [EXIGÊNCIA DE CÓDIGO 3c de 8]*

*# Função para consultar contatos*

*def consultar\_contatos():*

*while True:*

*opcao = input("Consultar (1-Todos / 2-Por Id / 3-Por Atividade / 4-Retornar): ")  # Pergunta qual opção deseja [EXIGÊNCIA DE CÓDIGO 4a de 8]*

*if opcao == "1":  # Consultar todos [EXIGÊNCIA DE CÓDIGO 4ai de 8]*

*for contato in lista\_contatos:*

*print(contato)*

*elif opcao == "2":  # Consultar por Id [EXIGÊNCIA DE CÓDIGO 4aii de 8]*

*try:*

*id\_pesquisa = int(input("Informe o id do contato: "))  # Certifique-se de que o ID é um número inteiro*

*encontrado = False*

*for contato in lista\_contatos:*

*if contato['id'] == id\_pesquisa:*

*print(contato)*

*encontrado = True*

*break*

*if not encontrado:*

*print("Id não encontrado.")*

*except ValueError:*

*print("Entrada inválida. O ID deve ser um número inteiro.")*

*elif opcao == "3":  # Consultar por Atividade [EXIGÊNCIA DE CÓDIGO 4aiii de 8]*

*atividade\_pesquisa = input("Informe a atividade: ")*

*encontrado = False*

*for contato in lista\_contatos:*

*if contato['atividade'] == atividade\_pesquisa:*

*print(contato)*

*encontrado = True*

*if not encontrado:*

*print("Nenhum contato encontrado com essa atividade.")*

*elif opcao == "4":  # Retornar ao menu [EXIGÊNCIA DE CÓDIGO 4aiv de 8]*

*return*

*else:*

*print("Opção inválida.")  # Opção inválida [EXIGÊNCIA DE CÓDIGO 4av de 8]*

*# Função para remover contatos*

*def remover\_contato():*

*while True:*

*try:*

*id\_remover = int(input("Informe o id do contato a ser removido: "))  # Pergunta o id do contato [EXIGÊNCIA DE CÓDIGO 5a de 8]*

*removido = False*

*for contato in lista\_contatos:*

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

*lista\_contatos.remove(contato)  # Remove o contato da lista [EXIGÊNCIA DE CÓDIGO 5b de 8]*

*removido = True*

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

*return*

*if not removido:*

*print("Id inválido.")  # Id inválido [EXIGÊNCIA DE CÓDIGO 5c de 8]*

*except ValueError:*

*print("Entrada inválida. Digite um número.")*

*# Menu principal [EXIGÊNCIA DE CÓDIGO 6 de 8]*

*while True:*

*opcao = input("Menu (1-Cadastrar / 2-Consultar / 3-Remover / 4-Encerrar): ")  # Pergunta qual opção deseja [EXIGÊNCIA DE CÓDIGO 6a de 8]*

*if opcao == "1":  # Cadastrar contato*

*id\_global += 1  # Incrementa o id\_global [EXIGÊNCIA DE CÓDIGO 6ai de 8]*

*cadastrar\_contato(id\_global)*

*elif opcao == "2":  # Consultar contato*

*consultar\_contatos()*

*elif opcao == "3":  # Remover contato*

*remover\_contato()*

*elif opcao == "4":  # Encerrar programa*

*print("Programa encerrado.")*

*break  # Sai do loop [EXIGÊNCIA DE CÓDIGO 6aiv de 8]*

*else:*

*print("Opção inválida.")  # Opção inválida [EXIGÊNCIA DE CÓDIGO 6av de 8]*

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

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM LARANJA PELA A SAÍDA DO CONSOLE DO EXERCÍCIO 4

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS

**SERÁ ACEITO SOMETE SAÍDAS DO CONSOLE NO FORMATO IMAGEM (NADA DE TEXTO AQUI! ZERA ESSA PARTE DA QUESTÃO!)**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente