QUESTÃO 1 de 4 - Conteúdo até Aula 03

Enunciado: Imagina-se que você é um dos programadores responsáveis pela construção de app de vendas para uma determinada empresa X que aceita cartões de crédito. Uma das estratégias de vendas dessa empresa X é cobrar um Juros maior conforme a quantidade de parcelas que o cliente desejar, conforme a **listagem abaixo**:

* Se a quantidade de parcelas for **menor** que **4**, o Juros será de **0%** (0 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **4** e **menor que** **6**, o Juros será de **4%** (4 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **6** e **menor que** **9**, o Juros será de **8%** (8 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **9** e **menor que** **13**, o Juros será de **16%** (16 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **13**, o Juros será de **32%** (32 / 100);

O valor da parcela é calculado da seguinte maneira:

O valor total parcelado é calculado da seguinte maneira:

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 loja do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 6];

1. Deve-se implementar o input do **valorDoPedido** e da **quantidadeParcelas** [EXIGÊNCIA DE CÓDIGO 2 de 6];
2. Deve-se implementar o Juros **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 **valorDaParcela** e **valorTotalParcelado** [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 um parcelamento com Juros (**quantidadeParcelas** maior ou igual a 4) apresentando o valor da Parcela e o valor Total Parcelado [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta a seguinte saída de código: 
Bem-vindo a Loja do Bruno Kostiuk
Entre com o valor do pedido: 1000.00
Entre com a quantidade de parcelas: 14
O valor das parcelas é de:R$ 94.29
O valor Total Parcelado é de:R$ 1320.00

**Figura 1.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o valor do pedido (pode ser qualquer valor) a quantidade de parcelas (maior ou igual a 4** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2]**), e é apresentado o valorDaParcela e o valorTotalParcelado.**

Apresentação de **Código da Questão 1**:

print ('Bem-vindos a loja da Gabriela Lopes!')

Valordopedido = float(input('Digite o valor do pedido: ')) # o comando float pode ser utilizado caso sejam números decimais

QuantidadeParcelas = int(input('Digite a quantidade de parcelas: ')) # o comando int deve ser utilizado somente para números inteiros

#Instruções que aplicam condições de juros maior conforme a quantidade de parcelas

if QuantidadeParcelas < 4:

  juros = 0/100

elif QuantidadeParcelas >= 4 and QuantidadeParcelas < 6:

  juros = 4/100

elif QuantidadeParcelas >= 6 and QuantidadeParcelas < 9:

  juros = 8/100

elif QuantidadeParcelas >= 9 and QuantidadeParcelas < 13:

  juros = 16/100

else:

  juros = 32/100

Valordaparcela = (Valordopedido \* (1 + juros) / QuantidadeParcelas)

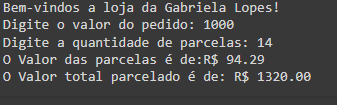
Valortotalparcelado = Valordaparcela \* QuantidadeParcelas

#Aqui estará os resultados dos calculos descritos acima

print ('O Valor das parcelas é de:R$ {0:.2f}'.format (Valordaparcela))

print ('O Valor total parcelado é de: R$ {0:.2f}'.format (Valortotalparcelado))

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 loja que vende Marmitas de Bife Acebolado ou Filé de Frango. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.

A Loja possui seguinte relação:

* Tamanho **P** de Bife Acebolado (**BA**) custa 16 reais e o Filé de Frango (**FF**) custa 15 reais;
* Tamanho **M** de Bife Acebolado (**BA**) custa 18 reais e o Filé de Frango (**FF**) custa 17 reais;
* Tamanho **G** de Bife Acebolado (**BA**) custa 22 reais e o Filé de Frango (**FF**) custa 21 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 loja de Marmitas 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** (BA/FF) e o print “Sabor inválido. Tente novamente" se o usuário entra com valor diferente de BA e FF [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 [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 a seguinte saída de código:

------ Bem-vindo a Loja de Marmitas do Bruno Kostiuk -----------
------------------------------Cardápio--------------------------
----------------------------------------------------------------
---| Tamanho  |  Bife Acebolado(BA)  |  Filé de Frango(FF)  |---
---|    P     |       R$ 16.00       |       R$ 15.00       |---
---|    M     |       R$ 18.00       |       R$ 17.00       |---
---|    G     |       R$ 22.00       |       R$ 21.00       |---
----------------------------------------------------------------
Entre com o sabor desejado (BA/FF): BF
Sabor inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): EXGG
Tamanho inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): M
Você pediu um Bife Acebolado no tamanho M: R$ 18.00

Deseja mais alguma coisa? (S/N): S
Entre com o sabor desejado (BA/FF): FF
Entre com o tamanho desejado (P/M/G): G
Você pediu um Filé de Frango no tamanho G: R$ 21.00

Deseja mais alguma coisa? (S/N): N

O valor total a ser pago: R$ 39.00

**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**:

print ('--------Bem-vindos a loja de Marmitas da Gabriela Lopes----------')

print ('---------------------------Cardápio------------------------------')

print ('-----------------------------------------------------------------')

print ('---|  Tamanho  |  Bife Acebolado(BA)  |  Filé de Frango(FF)  |---')

print ('---|     P     |       R$ 16.00       |       R$ 15.00       |---')

print ('---|     M     |       R$ 18.00       |       R$ 17.00       |---')

print ('---|     G     |       R$ 22.00       |       R$ 21.00       |---')

print ('-----------------------------------------------------------------')

acumulador = 0 #variavel que nos ajuda a somar os valores dos pedidos

while True:

  sabor = input ('Entre com o sabor desejado (BA/FF):')

  if sabor != 'BA' and sabor != 'FF':

    print ('Sabor invalido. Tente novamente!')

    continue  # Esse comando retorna ao inicio do laço se o usuario digitar o sabor errado

  tamanho = input ('Entre com o tamanho deseejado (P/M/G):')

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

    print ('tamanho invalido. Tente novamente!')

    continue  # Esse comando retorna ao inicio do laço se o usuario digitar o tamanho errado

# segue abaixo as combinações de sabor, tamanho e seus valores

  if sabor == 'BA' and tamanho == 'P':

    print ('Você pediu Bife Acebolado no tamanho P: R$ 16.00')

    acumulador += 16

  elif sabor == 'BA' and tamanho == 'M':

    print ('Você pediu Bife Acebolado no tamanho M: R$ 18.00')

    acumulador += 18

  elif sabor == 'BA' and tamanho == 'G':

    print ('Você pediu Bife Acebolado no tamanho G: R$ 22.00')

    acumulador += 22

  elif sabor == 'FF' and tamanho == 'P':

    print ('Você pediu Filé de Frango no tamanho P: R$ 15.00')

    acumulador += 15

  elif sabor == 'FF' and tamanho == 'M':

    print ('Você pediu Filé de Frango no tamanho M: R$ 17.00')

    acumulador += 17

  elif sabor == 'FF' and tamanho == 'G':

    print ('Você pediu Filé de Frango no tamanho G: R$ 21.00')

    acumulador += 21

  algomais = input ('Deseja pedir mais alguma coisa? (S/N):')

  if algomais == 'S':

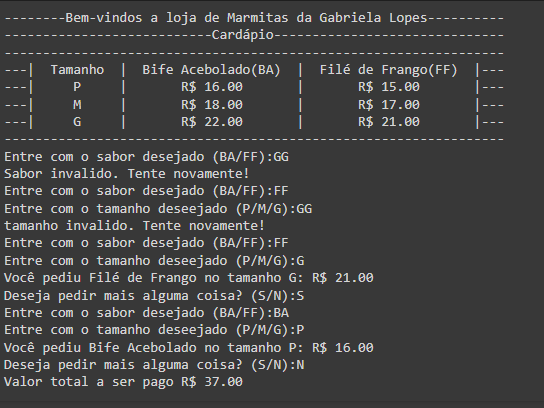
    continue # Esse comando retorna ao inicio do laço caso o usuario desejar contuniar fazendo pedidos

  elif algomais == 'N':

    print ('Valor total a ser pago R$ {0:.2f}' .format(acumulador))

    break # Esse comando para o laço de repetição caso o usuario desejar finalizar os pedidos

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 cobrança de serviços de uma fábrica que vende Camisetas em atacado. Você ficou com a parte de desenvolver a interface com o funcionário.

A Fábrica opera as vendas da seguinte maneira:

* Camiseta Manga Curta Simples (MCS), o valor unitário é de um real e oitenta centavos;
* Camiseta Manga Longa Simples (MLS), o valor unitário é de dois reais e dez centavos;
* Camiseta Manga Curta Com Estampa (MCE), o valor unitário é de dois reais e noventa centavos;
* Camiseta Manga Longa Com Estampa (MLE), o valor unitário é de três reais e vinte centavos;
* Se número de camisetas for **menor** que 20 não há desconto na venda;
* Se número de camisetas for **igual ou maior** que 20 e **menor** que 200, o desconto será de 5%;
* Se número de camisetas for **igual ou maior** que 200 e **menor** que 2000, o desconto será de 7%;
* Se número de camisetas for **igual ou maior** que 2000 e **menor ou igual** que 20000, o desconto será de 12%;
* Se número de camisetas for **maior** que 20000, não é aceito pedidos nessa quantidade de camisetas;

* Para o **adicional** de frete por transportadora (1) é cobrado um valor **extra** de 100 reais;
* Para o **adicional** de frete por Sedex (2) é cobrado um valor **extra** de 200 reais;
* Para o **adicional** de retirar o pedido na fábrica (0) é cobrado um valor **extra** de 0 reais;

O valor final da conta é calculado da seguinte maneira:

total = (**modelo \* num\_camisetas) + frete**

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 Fábrica de Camisetas do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 7];

1. Deve-se implementar a função **escolha\_modelo()** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Pergunta o **modelo** desejado;
   2. **Retorna** **o valor** **do modelo** com base na escolha do usuário (use **return**);
   3. Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
2. Deve-se implementar a função **num\_camisetas()** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Pergunta o **número de camisetas**;
   2. **Retorna** (use **return**) o **número de camisetas** com desconto seguindo a regra do enunciado (desconto calculado em cima do número de camisetas);
   3. Repete a pergunta do item **C.a** se digitar um valor acima de 20000 ou valor não numérico (use try/except para não numérico)
3. Deve-se implementar a função **frete()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Pergunta pelo serviço **adicional de frete**;
   2. **Retorna** (use **return**) o **valor** de apenas uma das **opções** de **frete**
   3. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/0;
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 modelo [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
9. Deve-se apresentar na saída de console um pedido no qual o usuário digitou ultrapassou no número de camisetas [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
10. Deve-se apresentar na saída de console um pedido com opção de modelo, número de camisetas e frete válidos [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta uma saída de console com o seguinte texto:

Bem vindo a Fábrica de Camisetas do do Bruno Kostiuk

Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MLL
Escolha inválida, entre com o modelo novamente


Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MCS
Entre com o número de camisetas: 300000
Não aceitamos tantas camisetas de uma vez.
Por favor, entre com o número de camisetas novamente.

Entre com o número de camisetas: 10000

Escolha o tipo de frete:
1 - Frete por transportadora - R$ 100.00
2 - Frete por Sedex - R$ 200.00
0 - Retirar pedido na fábrica - R$ 0.00
>>2
Total: R$ 16040.00 (Modelo: 1.80 * Quantidade(com desconto): 8800 + frete: 200.00)

**Figura 3.1: Exemplo de saída de console que o aluno deve fazer. Em que se pergunta pelo modelo e se erra opção inicialmente, que se passa do número de camisetas acima do aceito. Na sequência, o usuário digitou um modelo, número de camisetas e frete válidos.**

Apresentação de Código da Questão 3:

print('Bem-Vindos a fábrica de camisetas da Gabriela Lopes')

#função para definição dos valores do modelo

def escolha\_modelo():

   while True:

      modelo = input('Entre com o modelo desejado:\n'

                     '(MCS) - Manga Curta Simples '

                     '\n(MLS) - Manga Longa Simples'

                     '\n(MCE) - Manga Curta Com Estampa'

                     '\n(MLE) - Manga Longa Com Estampa\n'

                     '>>')

      if modelo == 'MCS':

         valor = 1.80

         return valor

      elif modelo == 'MLS':

         valor = 2.10

         return valor

      elif modelo == 'MCE':

         valor = 2.90

         return valor

      elif modelo == 'MLE':

         valor = 3.20

         return valor

      else:

         print("Opção inválida. Entre com o modelo novamente.")

#função para definição do desconto

def num\_camisetas():

   while True:

        try:

               numerocamisetas = int(input('                                 \n'

                                      "\nEntre com o número de camisetas: "))

        except :

               print('Opção invalida! Digite somente números inteiros!')

        if numerocamisetas > 20000:

         print('Não aceitamos tantas camisetas de uma vez.\n'

         'Por favor, entre com o númeo de camisetas novamente.')

         continue

        elif numerocamisetas < 20:

         return numerocamisetas

        elif 20 <= numerocamisetas < 200:

         return numerocamisetas \* (1 - 0.05)

        elif 200 <= numerocamisetas < 2000:

         return numerocamisetas \* (1 - 0.07)

        elif 2000 <= numerocamisetas <= 20000:

         return numerocamisetas \* (1 - 0.12)

#Função para definição do frete

def frete():

   while True:

       try:

          frete = int(input('                                \n\n'

                        'Escolha o tipo de frete: \n'

                        '1 - Frete por transportadora - R$ 100.00 \n'

                        '2 - Frete por Sedex - R$ 200.00 \n'

                        '0 - Retirar o pedido na fábrica - R$ 0.00 \n'

                        '>>'))

       except:

         print('Opção invalida. Digite somento números inteiros!')

       if frete == 0:

         frete = 0

         return frete

       elif frete == 1:

         frete = 100

         return frete

       elif frete == 2:

         frete = 200

         return frete

       continue

modelo = escolha\_modelo() #chamando função

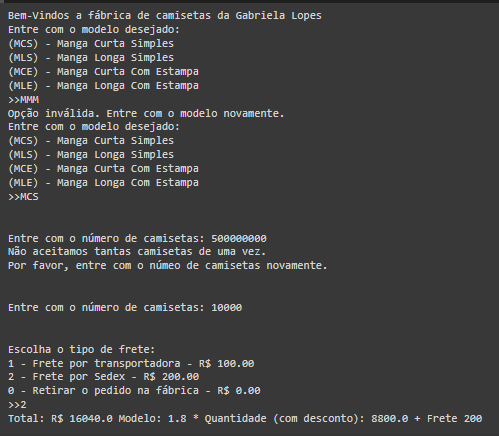
numerocamisetas = num\_camisetas() #chamando função

frete = frete() #chamando função

total = (modelo \* numerocamisetas) + frete

print(f"Total: R$ {total} Modelo: {modelo} \* Quantidade (com desconto): {numerocamisetas} + Frete {frete}")

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 pequena empresa para desenvolver o software de gerenciamento de funcionários. Este software deve ter o seguinte menu e opções:

1. Cadastrar Funcionário
2. Consultar Funcionário
   1. Consultar Todos
   2. Consultar por Id
   3. Consultar por setor
   4. Retornar ao menu
3. Remover Funcionário
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 empresa do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 8];

1. Deve-se implementar uma lista com o nome de **lista\_funcionarios** 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\_funcionario(id)** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 8];
   1. Pergunta **nome**, **setor**, **salario** do funcionário;
   2. Armazena o **id** (este é fornecido via parâmetro da função), **nome**, **setor**, **salario** dentro de um dicionário;
   3. **Copiar** o dicionário para dentro da **lista\_funcionarios** (utilizar o **copy**);
3. Deve-se implementar uma função chamada **consultar\_funcionarios()** em 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 funcionários com todos os seus dados cadastrados;
      2. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o funcionário específico com todos os seus dados cadastrados;
      3. Se Consultar por Setor, solicitar ao usuário que informe o setor, e apresentar o(s) funcionário(s) do setor 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 funcionários deve se repetir.
4. Deve-se implementar uma função chamada **remover\_funcionario()** em que**:** [EXIGÊNCIA DE CÓDIGO 5 de 8];
   1. Deve-se pergunta pelo **id** do funcionário a ser removido;
   2. Remover o funcionário da **lista\_funcionarios**;
   3. Se o id fornecido não for de um funcionário 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 Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
      1. Se Cadastrar Funcionário, **incrementar** em um **id\_ global** e chamar a função **cadastrar\_funcionario(id\_ global)**;
      2. Se Consultar Funcionário, chamar função **consultar\_funcionario ()**;
      3. Se Remover Funcionário, chamar função **remover\_funcionario()**;
      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 uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6];
9. Deve-se apresentar na saída de console um cadastro de 3 funcionários (sendo **2** deles no mesmo setor) [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 6];
10. Deve-se apresentar na saída de console uma consulta de todos os funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6];
11. Deve-se apresentar na saída de console uma consulta por código (id) de um dos funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6];
12. Deve-se apresentar na saída de console uma consulta por setor em que **2** funcionários sejam do mesmo setor [EXIGÊNCIA DE SAÍDA DE CONSOLE 5 de 6];
13. Deve-se apresentar na saída de console uma remoção de um dos funcionários seguida de uma consulta de todos os funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 6 de 6];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta o seguinte texto de saída de console: 

Bem vindo a Empresa do Bruno Kostiuk
--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297914
Por favor entre com o nome do Funcionário: Bruno
Por favor entre com o setor do Funcionário: Configurações
Por favor entre com o salário do Funcionário: 1000.00
--------------------------------------------------

**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 funcionário, note que o ID do funcionário inicia em 4297914 pois o id\_global é inicializado em 4297913.**

A imagem apresenta o seguinte texto de saída de console:

--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297915
Por favor entre com o nome do Funcionário: Tamy
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000
--------------------------------------------------

--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297916
Por favor entre com o nome do Funcionário: Osmar
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000.00
--------------------------------------------------

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

**A imagem apresenta o seguinte texto de saída de console:

--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>2
--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>1
----------------
id: 4297914
nome: Bruno
setor: Configurações
salário: 1000.0

id: 4297915
nome: Tamy
setor: Tutorias
salário: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salário: 1000.0

----------------**

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

A imagem apresenta o seguinte texto de saída de console: 

--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>2
Digite o id do funcionário: 4297914
----------------
id: 4297914
nome: Bruno
setor: Configurações
salario: 1000.0

----------------
--------------------------------------------------

--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>3
Digite o setor do(s) funcionário(s): Tutorias
----------------
id: 4297915
nome: Tamy
setor: Tutorias
salario: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salario: 1000.0

----------------

Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o funcionário com id número 4297914 e consulta pelo nome do setor (Tutorias).

A imagem apresenta o seguinte texto de saída do console:

--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>3
--------------------------------------------------
------------ MENU REMOVER FUNCIONÁRIO ------------------
Digite o id do funcionario a ser removido: 4297914
Funcionário removido com sucesso!
--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>2
--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>1
----------------
id: 4297915
nome: Tamy
setor: Tutorias
salário: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salário: 1000.0

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

Apresentação de **Código da Questão 4**:

print('Bem vindos a empresa da Gabriela Lopes')

lista\_funcionarios = [] #lista vazia para pôr cadastro funcionarios

id\_global = 4708574 #variavel para acumular os id

def cadastrar\_funcionario(id):#função cadastrar funcionarios

    global id\_global#chama id global

    print(40 \*'-')

    print(5 \* '-', 'MENU CADASTRAR FUNCIONÁRIOS', '-' \* 5)

    print('id do Funcionário:', id)

    nome = input('Por favor entre com nome do funcionário: ')

    setor  = input('Por favor entre com o setor do funcionário: ')

    while True:

        try:

            salario = float(input('Por favor entre com o salario do funcionario:'))

            break

        except ValueError:

          print('Valor inválido para salário. Tente novamente.')

    funcionario = {'id':id,'nome': nome,'setor':setor,'salario':salario} #atribuição de valores

    lista\_funcionarios.append(funcionario.copy()) #para adicionar a lista de funcionarios e copiar 'dicionario funcionario'

    id\_global += 1

def consultar\_funcionarios(): #função de consulta dos funcionarios e menu de opção

  while True:

      print(40 \*'-')

      print(5 \* '-', 'MENU CONSULTAR FUNCIONÁRIOS', '-' \* 5)

      opcao = input('Escolha uma opção do menu:\n'

      '1 - Consultar Todos os funcionários\n'

      '2 - Consultar funcionário por id\n'

      '3 - Consultar funcionário por setor\n'

      '4 - Retornar\n'

      '>>')

      if opcao == '1':       #opção de consulta todos

         for funcionario in lista\_funcionarios:

             print (f"id: {funcionario['id']}\n"

             f"Nome: {funcionario['nome']}\n"

             f"Setor: {funcionario['setor']}\n"

             f"Salario: {funcionario['salario']}")

      elif opcao == '2': #consulta por id

           id\_consulta = int(input('id do funcionario: '))

           for funcionario in lista\_funcionarios:

               if funcionario['id'] == id\_consulta:

                  print(f"ID: {funcionario['id']}\n"

                  f"Nome: {funcionario['nome']}\n"

                  f"Setor: {funcionario['setor']}\n"

                  f"Salario: {funcionario['salario']}")

                  break

           else:

               print('id invalido!')

      elif opcao == '3': #consulta por setor

           setor\_consulta = input('Setor do funcionário: ')

           for funcionario in lista\_funcionarios:

               if funcionario['setor'] == setor\_consulta:

                  print(f"ID: {funcionario['id']}\n"

                  f"Nome: {funcionario['nome']}\n"

                  f"Setor: {funcionario['setor']}\n"

                  f"Salario: {funcionario['salario']}")

      elif opcao == '4': #retornar ao MENU de opções

         return

      else: #caso usuario digite opção inválida

          print('Opção invalida!')

          continue

def remover\_funcionario(): #função que remove do cadastro do funcionario

    while True:

        id\_funcionario\_remover = int(input('Digite o id do funcionario a ser removido :  '))

        for funcionario in lista\_funcionarios: #consulta o id do funcionario e remove ele

            if funcionario['id'] == id\_funcionario\_remover:

                lista\_funcionarios. remove(funcionario)

                print(f'Funcionário removido com sucesso! ')

                return menuprincipal

        else: #mensagem de erro de ID inválido

            print('id invalido!')

            return

#menu principal

while True:

  print(40 \*'-')

  print(10 \* '-', 'MENU PRINCIPAL', '-' \* 10)

  menuprincipal = input('Escolha a opção desejada:\n'

  '1 - Cadastrar Funcionário \n'

  '2 - Consultar Funcionário \n'

  '3 - Remover Funcionário \n'

  '4 - Sair \n'

  '>>')

  if menuprincipal == '1': #para cadastrar

     print('')

     cadastrar\_funcionario(id\_global + 1)

  elif menuprincipal == '2': #para consulta

       print('')

       consultar\_funcionarios()

  elif menuprincipal == '3':  #para remover

       print('')

       remover\_funcionario()

  elif menuprincipal == '4': #para encerrar o programa - encerra o loops e printa mensagem

       print('')

       break

  else: #para opção invalida na entrada do menu

      print('Opção inválida')

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

