



UNINTER

ATIVIDADE PRÁTICA MÓDULO B LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

Allana Caroline Philip Alves RU: 4907817

Prof. Me. Bruno Kostiuk

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

O valor da parcela é calculado da seguinte maneira:

$$\text{valorDaParcela} = \frac{\text{valorDoPedido} * (1 + \text{juros})}{\text{quantidadeParcelas}}$$

O valor total parcelado é calculado da seguinte maneira:

$$\text{valorTotalParcelado} = \text{valorDaParcela} * \text{quantidadeParcelas}$$

Elabore um programa em Python que:

- Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). [EXIGÊNCIA DE CÓDIGO 1 de 6];
- Deve-se implementar o input do **valorDoPedido** e da **quantidadeParcelas** [EXIGÊNCIA DE CÓDIGO 2 de 6];
- 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];
- Deve-se implementar o **valorDaParcela** e **valorTotalParcelado** [EXIGÊNCIA DE CÓDIGO 4 de 6];
- Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
- Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 6 de 6];

Teste seu código atendendo as seguintes exigências:

- Deve-se apresentar na saída de console uma mensagem com seu nome e sobrenome [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];
- 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];

Apresentação de Código da Questão 1:

```
# Mensagem de boas-vindas

print("Seja bem-vindo à loja da Allana Alves!")


# Entrada de dados do usuário

valorDoPedido = float(input("Digite o valor do pedido: "))
quantidadeParcelas = int(input("Digite a quantidade de parcelas: "))


# Definindo a taxa de juros com base na quantidade de parcelas

if quantidadeParcelas < 4:

    juros = 0    # Sem juros

elif quantidadeParcelas >= 4 and quantidadeParcelas < 6:

    juros = 0.04    # 4% de juros

elif quantidadeParcelas >= 6 and quantidadeParcelas < 9:

    juros = 0.08    # 8% de juros

elif quantidadeParcelas >= 9 and quantidadeParcelas < 13:

    juros = 0.16    # 16% de juros

else:

    juros = 0.32    # 32% de juros


# Cálculo do valor da parcela

valorDaParcela = (valorDoPedido * (1 + juros)) / quantidadeParcelas


# Cálculo do valor total parcelado

valorTotalParcelado = valorDaParcela * quantidadeParcelas


# Exibindo os resultados

print("Programador: Allana Alves")    # Nome e sobrenome


# Exibindo valores apenas se tiver juros (ou seja, parcelas a partir de 4)

if quantidadeParcelas >= 4:

    print(f"Valor da parcela: R${valorDaParcela:.2f}")

    print(f"Valor total parcelado: R${valorTotalParcelado:.2f}")

else:
```

```
print("Compra realizada sem juros.")  
print(f"Valor da parcela: R${valorDaParcela:.2f}")  
print(f"Valor total: R${valorTotalParcelado:.2f}")
```

Apresentação de Saída do Console da Questão 1:

```
... Seja bem-vindo à loja da Allana Alves!  
    Digite o valor do pedido: 
```

```
... Seja bem-vindo à loja da Allana Alves!  
    Digite o valor do pedido: 1500.00  
    Digite a quantidade de parcelas: 
```

```
⇒ Seja bem-vindo à loja da Allana Alves!  
   Digite o valor do pedido: 1500.00  
   Digite a quantidade de parcelas: 10  
   Programador: Allana Alves  
   Valor da parcela: R$174.00  
   Valor total parcelado: R$1740.00
```

QUESTÃO 2 de 4 - Conteúdo até aula 04

Enunciado: Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma 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:

- Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). Além do seu nome e sobrenome, deve-se implementar um **print com um Menu** para o cliente. [EXIGÊNCIA DE CÓDIGO 1 de 8];
- 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];
- 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];
- 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];
- Deve-se implementar um **acumulador** para somar os valores dos pedidos [EXIGÊNCIA DE CÓDIGO 5 de 8];
- 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];
- Deve-se implementar as estruturas de **while, break, continue (todas elas)** [EXIGÊNCIA DE CÓDIGO 7 de 8];
- Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 8 de 8];

Teste seu código atendendo as seguintes exigências:

- 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];
- Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
- Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
- 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];

Apresentação de Código da Questão 2:

```
# EXIGÊNCIA A – Mensagem de boas-vindas com nome e sobrenome
print("Bem-vindo à loja de Marmitas da Allana Alves!")
```

```
# EXIGÊNCIA I – Nome completo na apresentação do menu
print("Atendimento realizado por: Allana Caroline Philip Alves\n")
```

```
# EXIBIÇÃO DO MENU
```

```

print("Menu de Marmitas:")
print("Tamanho P - Bife Acebolado (BA): R$16,00 | Filé de Frango (FF): R$15,00")
print("Tamanho M - Bife Acebolado (BA): R$18,00 | Filé de Frango (FF): R$17,00")
print("Tamanho G - Bife Acebolado (BA): R$22,00 | Filé de Frango (FF): R$21,00\n")

# Inicializa o acumulador para somar os pedidos
total_pedido = 0.0

# Laço principal para repetição do pedido
while True:
    # Entrada do sabor
    sabor = input("Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): ").upper()
    if sabor != "BA" and sabor != "FF":
        print("Sabor inválido. Tente novamente.\n") # EXIGÊNCIA J
        continue # Volta para o início do laço se o sabor estiver incorreto

    # Entrada do tamanho
    tamanho = input("Digite o tamanho desejado (P, M ou G): ").upper()
    if tamanho != "P" and tamanho != "M" and tamanho != "G":
        print("Tamanho inválido. Tente novamente.\n") # EXIGÊNCIA K
        continue # Volta para o início do laço se o tamanho estiver incorreto

    # Verifica o valor com base no sabor e tamanho (if aninhado)
    if sabor == "BA": # Bife Acebolado
        if tamanho == "P":
            preco = 16.0
        elif tamanho == "M":
            preco = 18.0
        else: # tamanho == "G"
            preco = 22.0
    elif sabor == "FF": # Filé de Frango
        if tamanho == "P":
            preco = 15.0
        elif tamanho == "M":
            preco = 17.0
        else: # tamanho == "G"
            preco = 21.0

    # Mostra o valor do item escolhido
    print(f"Você escolheu uma marmita de {sabor} tamanho {tamanho}. Valor: R${preco:.2f}\n")

    # Soma ao total acumulado (EXIGÊNCIA E)
    total_pedido += preco

    # Pergunta se deseja continuar (EXIGÊNCIA F)
    continuar = input("Deseja pedir mais alguma coisa? (s/n): ").lower()
    if continuar == "s":
        continue # EXIGÊNCIA G - uso de continue
    else:
        break # EXIGÊNCIA G - uso de break

# EXIBIÇÃO FINAL DO TOTAL
print(f"\nValor total do seu pedido: R${total_pedido:.2f}")
print("Obrigada pela preferência!")

```

Apresentação de Saída do Console da Questão 2:

*** Bem-vindo à loja de Marmitas da Allana Alves!
Atendimento realizado por: Allana Caroline Philip Alves

Menu de Marmitas:
Tamanho P - Bife Acebolado (BA): R\$16,00 | Filé de Frango (FF): R\$15,00
Tamanho M - Bife Acebolado (BA): R\$18,00 | Filé de Frango (FF): R\$17,00
Tamanho G - Bife Acebolado (BA): R\$22,00 | Filé de Frango (FF): R\$21,00

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango):

*** Bem-vindo à loja de Marmitas da Allana Alves!
Atendimento realizado por: Allana Caroline Philip Alves

Menu de Marmitas:
Tamanho P - Bife Acebolado (BA): R\$16,00 | Filé de Frango (FF): R\$15,00
Tamanho M - Bife Acebolado (BA): R\$18,00 | Filé de Frango (FF): R\$17,00
Tamanho G - Bife Acebolado (BA): R\$22,00 | Filé de Frango (FF): R\$21,00

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): DD
Sabor inválido. Tente novamente.

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango):



Bem-vindo à loja de Marmitas da Allana Alves!
Atendimento realizado por: Allana Caroline Philip Alves

Menu de Marmitas:

Tamanho P - Bife Acebolado (BA): R\$16,00 | Filé de Frango (FF): R\$15,00
Tamanho M - Bife Acebolado (BA): R\$18,00 | Filé de Frango (FF): R\$17,00
Tamanho G - Bife Acebolado (BA): R\$22,00 | Filé de Frango (FF): R\$21,00

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): DD
Sabor inválido. Tente novamente.

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): BA
Digite o tamanho desejado (P, M ou G): BIG
Tamanho inválido. Tente novamente.

Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): BA
Digite o tamanho desejado (P, M ou G): G
Você escolheu uma marmita de BA tamanho G. Valor: R\$22.00

Deseja pedir mais alguma coisa? (s/n): S
Digite o sabor desejado (BA para Bife Acebolado ou FF para Filé de Frango): FF
Digite o tamanho desejado (P, M ou G): P
Você escolheu uma marmita de FF tamanho P. Valor: R\$15.00

Deseja pedir mais alguma coisa? (s/n): N

Valor total do seu pedido: R\$37.00
Obrigada pela preferência!

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:

$$\text{total} = (\text{modelo} * \text{num_camisetas}) + \text{frete}$$

Elabore um programa em Python que:

- A. Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). [EXIGÊNCIA DE CÓDIGO 1 de 7];
 - B. Deve-se implementar a função **escolha_modelo()** em que: [EXIGÊNCIA DE CÓDIGO 2 de 7];
 - a. Pergunta o **modelo** desejado;
 - b. **Retorna o valor do modelo** com base na escolha do usuário (use **return**);
 - c. Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
 - C. Deve-se implementar a função **num_camisetas()** em que: [EXIGÊNCIA DE CÓDIGO 3 de 7];
 - a. Pergunta o **número de camisetas**;
 - b. **Retorna** (use **return**) o **número de camisetas** com desconto seguindo a regra do enunciado (desconto calculado em cima do número de camisetas);
 - c. Repete a pergunta do item **C.a** se digitar um valor acima de 20000 ou valor não numérico (use try/except para não numérico)
 - D. Deve-se implementar a função **frete()** em que: [EXIGÊNCIA DE CÓDIGO 4 de 7];
 - a. Pergunta pelo serviço **adicional de frete**;
 - b. **Retorna** (use **return**) o **valor** de apenas uma das **opções de frete**
 - c. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/0;
 - E. Deve-se implementar o total a pagar no código principal (**main**), ou seja, **não pode estar dentro de função**, conforme o enunciado [EXIGÊNCIA DE CÓDIGO 5 de 7];
 - F. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
 - G. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
- Teste seu código atendendo as seguintes exigências:**
- H. Deve-se apresentar na saída de console uma mensagem com o seu **nome e sobrenome** [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
 - I. Deve-se apresentar na saída de console um pedido no qual o usuário errou a opção de modelo [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
 - J. 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];
 - K. 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];

Apresentação de Código da Questão 3:

```
# EXIGÊNCIA A e H - Mensagem de boas-vindas com nome e sobrenome
print("Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves\n")

# Explicando o que significa cada sigla dos modelos
print("Modelos disponíveis:")

print("MCS - Camiseta Manga Curta Simples - R$1,80")
print("MLS - Camiseta Manga Longa Simples - R$2,10")
print("MCE - Camiseta Manga Curta com Estampa - R$2,90")
print("MLE - Camiseta Manga Longa com Estampa - R$3,20\n")

# Função para escolher o modelo da camiseta
```

```

def escolha_modelo():
    while True:
        modelo = input("Digite o modelo desejado (MCS, MLS, MCE, MLE): ").upper()
        # Valida e retorna o valor unitário conforme o modelo
        if modelo == "MCS":
            return 1.80
        elif modelo == "MLS":
            return 2.10
        elif modelo == "MCE":
            return 2.90
        elif modelo == "MLE":
            return 3.20
        else:
            print("Modelo inválido. Tente novamente.") # EXIGÊNCIA I

# Função para digitar a quantidade de camisetas com regras de desconto
def num_camisetas():
    while True:
        try:
            quantidade = int(input("Digite o número de camisetas: "))
            if quantidade > 20000:
                print("Quantidade não permitida. Máximo permitido: 20000.") # EXIGÊNCIA J
                continue
            elif quantidade >= 2000:
                return quantidade * 0.88 # 12% de desconto
            elif quantidade >= 200:
                return quantidade * 0.93 # 7% de desconto
            elif quantidade >= 20:
                return quantidade * 0.95 # 5% de desconto
            elif quantidade > 0:
                return quantidade * 1.0 # Sem desconto
            else:
                print("Digite um número maior que zero.")
        except:
            print("Valor inválido. Digite apenas números.") # EXIGÊNCIA F

# Função para escolher o tipo de frete
def frete():

```

```

while True:

    print("\nOpções de frete:")
    print("0 - Retirar na fábrica (frete grátis)")
    print("1 - Transportadora (+R$100,00)")
    print("2 - Sedex (+R$200,00)")
    opcao = input("Escolha o tipo de frete (0/1/2): ")
    if opcao == "0":
        return 0
    elif opcao == "1":
        return 100
    elif opcao == "2":
        return 200
    else:
        print("Opção de frete inválida. Tente novamente.")

# CÓDIGO PRINCIPAL - EXIGÊNCIA E
preco_unitario = escolha_modelo()          # Escolha e validação do modelo
quantidade_final = num_camisetas()         # Quantidade com desconto aplicado
valor_frete = frete()                      # Escolha do frete

# Cálculo do total
total = (preco_unitario * quantidade_final) + valor_frete

# Exibição do valor total
print(f"\nTotal a pagar: R${total:.2f}")
print("Pedido registrado com sucesso. Obrigado por utilizar o sistema!")

```

Apresentação de Saída do Console da Questão 3:

... Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves

Modelos disponíveis:

MCS - Camiseta Manga Curta Simples - R\$1,80

MLS - Camiseta Manga Longa Simples - R\$2,10

MCE - Camiseta Manga Curta com Estampa - R\$2,90

MLE - Camiseta Manga Longa com Estampa - R\$3,20

Digite o modelo desejado (MCS, MLS, MCE, MLE):

... Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves

Modelos disponíveis:

MCS - Camiseta Manga Curta Simples - R\$1,80

MLS - Camiseta Manga Longa Simples - R\$2,10

MCE - Camiseta Manga Curta com Estampa - R\$2,90

MLE - Camiseta Manga Longa com Estampa - R\$3,20

Digite o modelo desejado (MCS, MLS, MCE, MLE): XXE

Modelo inválido. Tente novamente.

Digite o modelo desejado (MCS, MLS, MCE, MLE):

... Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves

Modelos disponíveis:

MCS - Camiseta Manga Curta Simples - R\$1,80

MLS - Camiseta Manga Longa Simples - R\$2,10

MCE - Camiseta Manga Curta com Estampa - R\$2,90

MLE - Camiseta Manga Longa com Estampa - R\$3,20

Digite o modelo desejado (MCS, MLS, MCE, MLE): XXE

Modelo inválido. Tente novamente.

Digite o modelo desejado (MCS, MLS, MCE, MLE): MCS

Digite o número de camisetas: 561387951648

Quantidade não permitida. Máximo permitido: 20000.

Digite o número de camisetas:

*** Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves

Modelos disponíveis:

MCS - Camiseta Manga Curta Simples - R\$1,80

MLS - Camiseta Manga Longa Simples - R\$2,10

MCE - Camiseta Manga Curta com Estampa - R\$2,90

MLE - Camiseta Manga Longa com Estampa - R\$3,20

Digite o modelo desejado (MCS, MLS, MCE, MLE): XXE

Modelo inválido. Tente novamente.

Digite o modelo desejado (MCS, MLS, MCE, MLE): MCS

Digite o número de camisetas: 561387951648

Quantidade não permitida. Máximo permitido: 20000.

Digite o número de camisetas: 900

Opções de frete:

0 - Retirar na fábrica (frete grátis)

1 - Transportadora (+R\$100,00)

2 - Sedex (+R\$200,00)

Escolha o tipo de frete (0/1/2):



Bem-vindo ao sistema de pedidos da fábrica de camisetas! Atendente: Allana Alves

Modelos disponíveis:

MCS - Camiseta Manga Curta Simples - R\$1,80

MLS - Camiseta Manga Longa Simples - R\$2,10

MCE - Camiseta Manga Curta com Estampa - R\$2,90

MLE - Camiseta Manga Longa com Estampa - R\$3,20

Digite o modelo desejado (MCS, MLS, MCE, MLE): XXE

Modelo inválido. Tente novamente.

Digite o modelo desejado (MCS, MLS, MCE, MLE): MCS

Digite o número de camisetas: 561387951648

Quantidade não permitida. Máximo permitido: 20000.

Digite o número de camisetas: 900

Opções de frete:

0 - Retirar na fábrica (frete grátis)

1 - Transportadora (+R\$100,00)

2 - Sedex (+R\$200,00)

Escolha o tipo de frete (0/1/2): 1

Total a pagar: R\$1606.60

Pedido registrado com sucesso. Obrigado por utilizar o sistema!

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:

- A. Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui) [EXIGÊNCIA DE CÓDIGO 1 de 8];
- B. 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];
- C. Deve-se implementar uma função chamada **cadastrar_funcionario(id)** em que: [EXIGÊNCIA DE CÓDIGO 3 de 8];
 - a. Pergunta **nome, setor, salario** do funcionário;
 - b. Armazena o **id** (este é fornecido via parâmetro da função), **nome, setor, salario** dentro de um dicionário;
 - c. **Copiar** o dicionário para dentro da **lista_funcionarios**, usando **.copy()** ou similar;
- D. Deve-se implementar uma função chamada **consultar_funcionarios()** em que: [EXIGÊNCIA DE CÓDIGO 4 de 8];
 - a. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
 - i. Se Consultar Todos, apresentar todos os funcionários com todos os seus dados cadastrados;
 - ii. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o funcionário específico com todos os seus dados cadastrados;
 - iii. 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;
 - iv. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
 - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **D.a.**
 - vi. Enquanto o usuário não escolher a opção 4, o menu consultar funcionários deve se repetir.
- E. Deve-se implementar uma função chamada **remover_funcionario()** em que: [EXIGÊNCIA DE CÓDIGO 5 de 8];
 - a. Deve-se pergunta pelo **id** do funcionário a ser removido;
 - b. Remover o funcionário da **lista_funcionarios**;
 - c. Se o id fornecido não for de um funcionário da lista, printar "Id inválido" e repetir a pergunta **E.a.**

- F. Deve-se implementar uma estrutura de menu no código principal (**main**), ou seja, **não pode estar dentro de função**, em que: **[EXIGÊNCIA DE CÓDIGO 6 de 8]**:
- a. Deve-se perguntar qual opção deseja (1. Cadastrar Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
 - i. Se Cadastrar Funcionário, chamar a função **cadaststrar_funcionario(id_ global)** e **em seguida, incrementar** em um **id_ global** (no menu principal);
 - ii. Se Consultar Funcionário, chamar função **consultar_funcionario ()**;
 - iii. Se Remover Funcionário, chamar função **remover_funcionario()**;
 - iv. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
 - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **F.a.**
 - vi. Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
- G. Deve-se implementar uma **lista de dicionários** (uma lista contendo dicionários dentro) **[EXIGÊNCIA DE CÓDIGO 7 de 8]**:
- H. Deve-se inserir comentários relevantes no código **[EXIGÊNCIA DE CÓDIGO 8 de 8]**:

Teste seu código atendendo as seguintes exigências:

- I. Deve-se apresentar na saída de console uma mensagem com o seu **nome e sobrenome** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6]**;
- J. 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]**;
- K. Deve-se apresentar na saída de console uma **consulta de todos os funcionários** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6]**;
- L. Deve-se apresentar na saída de console uma **consulta por código (id)** de um dos funcionários **[EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6]**;
- M. 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]**;
- N. 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]**;

Apresentação de *Código da Questão 4*:

```
# EXIGÊNCIA A e I - Mensagem de boas-vindas com nome e sobrenome

print("Bem-vindo ao sistema de gerenciamento de funcionários! Desenvolvido por Allana
Alves.\n")


# EXIGÊNCIA B - Lista para armazenar funcionários e ID inicial (use seu RU real se tiver;
aqui usarei um exemplo)

lista_funcionarios = []

id_global = 1234567 # substitua pelo seu RU se desejar


# EXIGÊNCIA C - Função para cadastrar funcionário

def cadastrar_funcionario(id):
    nome = input("Digite o nome do funcionário: ")
    setor = input("Digite o setor do funcionário: ")
    salario = input("Digite o salário do funcionário: ")

    funcionario = {
        "id": id,
        "nome": nome,
        "setor": setor,
        "salario": salario
    }

    lista_funcionarios.append(funcionario.copy()) # Adiciona uma cópia do dicionário à
lista


# EXIGÊNCIA D - Função para consultar funcionários

def consultar_funcionarios():
    while True:
        print("\nConsulta de Funcionários:")
        print("1 - Consultar Todos")
        print("2 - Consultar por Id")
        print("3 - Consultar por Setor")
        print("4 - Retornar ao menu")
        opcao = input("Escolha uma opção: ")

        if opcao == "1":
            # EXIGÊNCIA K - Mostrar todos os funcionários
            for func in lista_funcionarios:
```



```

        print(func)
elif opcao == "2":
    # EXIGÊNCIA L - Consultar por ID
    id_busca = input("Digite o ID do funcionário: ")
    encontrado = False
    for func in lista_funcionarios:
        if str(func["id"]) == id_busca:
            print(func)
            encontrado = True
    if not encontrado:
        print("Funcionário não encontrado.")
elif opcao == "3":
    # EXIGÊNCIA M - Consultar por setor
    setor_busca = input("Digite o setor: ")
    encontrado = False
    for func in lista_funcionarios:
        if func["setor"].lower() == setor_busca.lower():
            print(func)
            encontrado = True
    if not encontrado:
        print("Nenhum funcionário encontrado neste setor.")
elif opcao == "4":
    return # Volta para o menu principal
else:
    print("Opção inválida. Tente novamente.")

```

EXIGÊNCIA E - Função para remover funcionário

```

def remover_funcionario():
    while True:
        id_remove = input("Digite o ID do funcionário que deseja remover: ")
        encontrado = False
        for func in lista_funcionarios:
            if str(func["id"]) == id_remove:
                lista_funcionarios.remove(func)
                print("Funcionário removido com sucesso.")
                return
        print("ID inválido. Tente novamente.")

```

```
# EXIGÊNCIA F - Menu principal
while True:
    print("\nMenu Principal:")
    print("1 - Cadastrar Funcionário")
    print("2 - Consultar Funcionário")
    print("3 - Remover Funcionário")
    print("4 - Encerrar Programa")
    escolha = input("Escolha uma opção: ")

    if escolha == "1":
        cadastrar_funcionario(id_global) # Chama função de cadastro
        id_global += 1 # Incrementa ID
    elif escolha == "2":
        consultar_funcionarios() # Chama função de consulta
    elif escolha == "3":
        remover_funcionario() # Chama função de remoção
    elif escolha == "4":
        print("Programa encerrado. Até logo!")
        break # Encerra o programa
    else:
        print("Opção inválida. Tente novamente.")
```

Apresentação de Saída do Console da Questão 4

```
*** Bem-vindo ao sistema de gerenciamento de funcionários! Desenvolvido por Allana Alves.
```

```
Menu Principal:
```

```
1 - Cadastrar Funcionário
```

```
2 - Consultar Funcionário
```

```
3 - Remover Funcionário
```

```
4 - Encerrar Programa
```

```
Escolha uma opção: 
```

*** Bem-vindo ao sistema de gerenciamento de funcionários! Desenvolvido por Allana Alves.

Menu Principal:

- 1 - Cadastrar Funcionário
- 2 - Consultar Funcionário
- 3 - Remover Funcionário
- 4 - Encerrar Programa

Escolha uma opção: 1

Digite o nome do funcionário: João da Silva

Digite o setor do funcionário: Financeiro

Digite o salário do funcionário: 3000

Menu Principal:

- 1 - Cadastrar Funcionário
- 2 - Consultar Funcionário
- 3 - Remover Funcionário
- 4 - Encerrar Programa

Escolha uma opção:

Menu Principal:

```
... 1 - Cadastrar Funcionário
     2 - Consultar Funcionário
     3 - Remover Funcionário
     4 - Encerrar Programa
Escolha uma opção: 1
Digite o nome do funcionário: Ana Souza
Digite o setor do funcionário: RH
Digite o salário do funcionário: 2800
```

Menu Principal:

```
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 1
Digite o nome do funcionário: Carlos Lima
Digite o setor do funcionário: RH
Digite o salário do funcionário: 2900
```

Menu Principal:

```
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 
```

```
Digite o nome do funcionário: Carlos Lima
Digite o setor do funcionário: RH
... Digite o salário do funcionário: 2900
```

Menu Principal:

```
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 2
```

Consulta de Funcionários:

```
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 1
{'id': 1234567, 'nome': 'João da Silva', 'setor': 'Financeiro', 'salario': '3000'}
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

Consulta de Funcionários:

```
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 
```

```
Consulta de Funcionários:
... 1 - Consultar Todos
     2 - Consultar por Id
     3 - Consultar por Setor
     4 - Retornar ao menu
Escolha uma opção: 1
{'id': 1234567, 'nome': 'João da Silva', 'setor': 'Financeiro', 'salario': '3000'}
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 2
Digite o ID do funcionário: 1234567
{'id': 1234567, 'nome': 'João da Silva', 'setor': 'Financeiro', 'salario': '3000'}
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 3
Digite o setor: RH
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 
```

Consulta de Funcionários:

- 1 - Consultar Todos
- 2 - Consultar por Id
- 3 - Consultar por Setor
- 4 - Retornar ao menu

Escolha uma opção: 3

Digite o setor: RH

```
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
```

```
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

Consulta de Funcionários:

- 1 - Consultar Todos
 - 2 - Consultar por Id
 - 3 - Consultar por Setor
 - 4 - Retornar ao menu
- Escolha uma opção: 4

Menu Principal:

- 1 - Cadastrar Funcionário
- 2 - Consultar Funcionário
- 3 - Remover Funcionário
- 4 - Encerrar Programa

Escolha uma opção:

```
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
*** 4 - Retornar ao menu
Escolha uma opção: 4

Menu Principal:
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 3
Digite o ID do funcionário que deseja remover: 1234567
Funcionário removido com sucesso.
```

```
Menu Principal:
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 
```

```
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
*** 4 - Encerrar Programa
Escolha uma opção: 2

Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 1
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 
```

```
2 - Consultar por Id
3 - Consultar por Setor
*** 4 - Retornar ao menu
Escolha uma opção: 1
{'id': 1234568, 'nome': 'Ana Souza', 'setor': 'RH', 'salario': '2800'}
{'id': 1234569, 'nome': 'Carlos Lima', 'setor': 'RH', 'salario': '2900'}
```

```
Consulta de Funcionários:
1 - Consultar Todos
2 - Consultar por Id
3 - Consultar por Setor
4 - Retornar ao menu
Escolha uma opção: 4
```

```
Menu Principal:
1 - Cadastrar Funcionário
2 - Consultar Funcionário
3 - Remover Funcionário
4 - Encerrar Programa
Escolha uma opção: 
```