QUESTÃO 1 de 4 - Conteúdo até **aula 03**

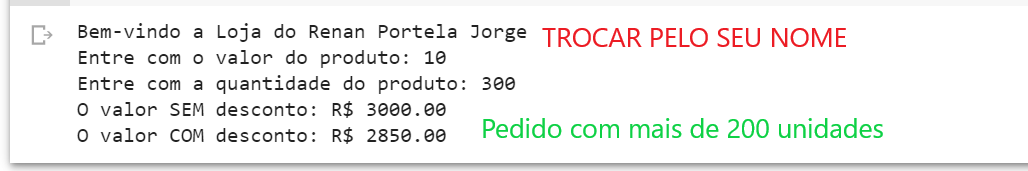
Enunciado: Imagina-se que você é um dos programadores responsáveis pela construção de app de vendas para uma determinada empresa X que vende em atacado. Uma das estratégias de vendas dessa empresa X é dar desconto maiores por unidade as informações abaixo:

* Se quantidade for menor que 200 o desconto será de 0%;
* Se quantidade for igual ou maior que 200 e menor que 1000 o desconto será de 5%;
* Se quantidade for igual ou maior que1000 e menor que 2000 o desconto será de 10%;
* Se quantidade for igual ou maior que 2000 o desconto será de 15%;

Elabore um programa em Python que:

1. Realizar o print uma mensagem de boas-vindas que apareça o seu nome;
2. Deve-se entrar com o **valor unitário** e **quantidade** do produto [EXIGÊNCIA DE CÓDIGO 1 de 4];
3. Deve-se retornar o **valor total sem desconto** e o **valor total com desconto**[EXIGÊNCIA DE CÓDIGO 2 de 4];
4. Deve-se utilizar as estruturas **if, elif e else (todas elas)**[EXIGÊNCIA DE CÓDIGO 3 de 4];
5. Deve-se fazer comentários no código [EXIGÊNCIA DE CÓDIGO 4 de 4];
6. Deve-se colocar na apresentação de saída de consoleum pedido recebendo desconto[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 1];

EXEMPLO DE SAÍDA DE CONSOLE:



**Figura 1: Exemplo de saída de console que o aluno deve fazer.Em que se perguntar o valor do produto (pode ser qualquer valor) a quantidade (deve ser maior que 200) e apresenta o valor final sem o desconto e com o desconto.**

Código da Questão 1:

# A. Mensagem de boas-vindas abaixo

nome = "Gabriel Rodrigues de Almeida Messa"

print(f"Bem-vindo, {nome}!")

# B. Entrada de valor e quantidade do produto

valor\_unitario\_do\_produto = float(input("Digite o valor unitário do produto: "))

quantidades = int(input("Digite a quantidade do produto: "))

# C. Cálculo do valor total sem desconto

valor\_total\_sem\_desconto = valor\_unitario\_do\_produto \* quantidades

# Cálculo do desconto com base na quantidade

if quantidades < 200:

desconto\_percentual = 0

elif quantidades < 1000:

desconto\_percentual = 0.05

elif quantidades < 2000:

desconto\_percentual = 0.1

else:

desconto\_percentual = 0.15

# Cálculo do valor total com desconto

valor\_desconto = valor\_total\_sem\_desconto \* desconto\_percentual

valor\_total\_com\_desconto = valor\_total\_sem\_desconto - valor\_desconto

# D. Exibição do valor total sem desconto e com desconto

print(f"Valor total sem desconto: R${valor\_total\_sem\_desconto:.2f}")

print(f"Desconto aplicado: {desconto\_percentual \* 100:.0f}%")

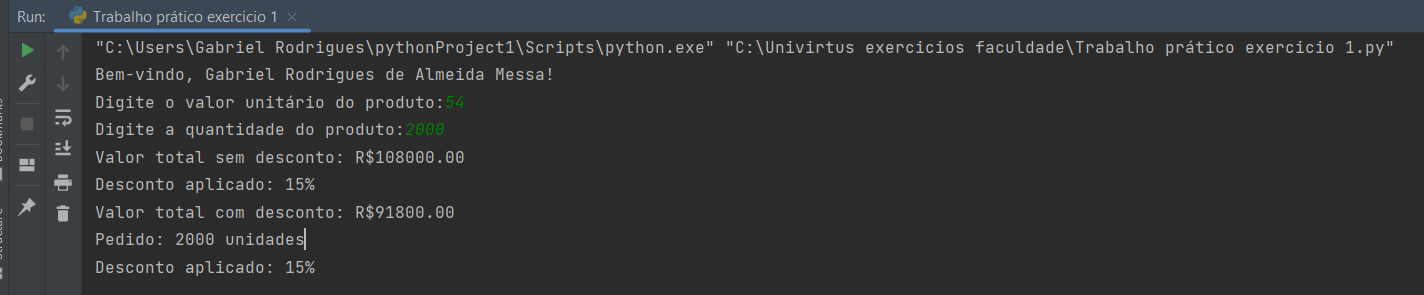
print(f"Valor total com desconto: R${valor\_total\_com\_desconto:.2f}")

# Exibe a saída do pedido e o desconto aplicado

print(f"Pedido: {quantidades} unidades")

print(f"Desconto aplicado: {desconto\_percentual \* 100:.0f}%")

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 sorveteria. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.

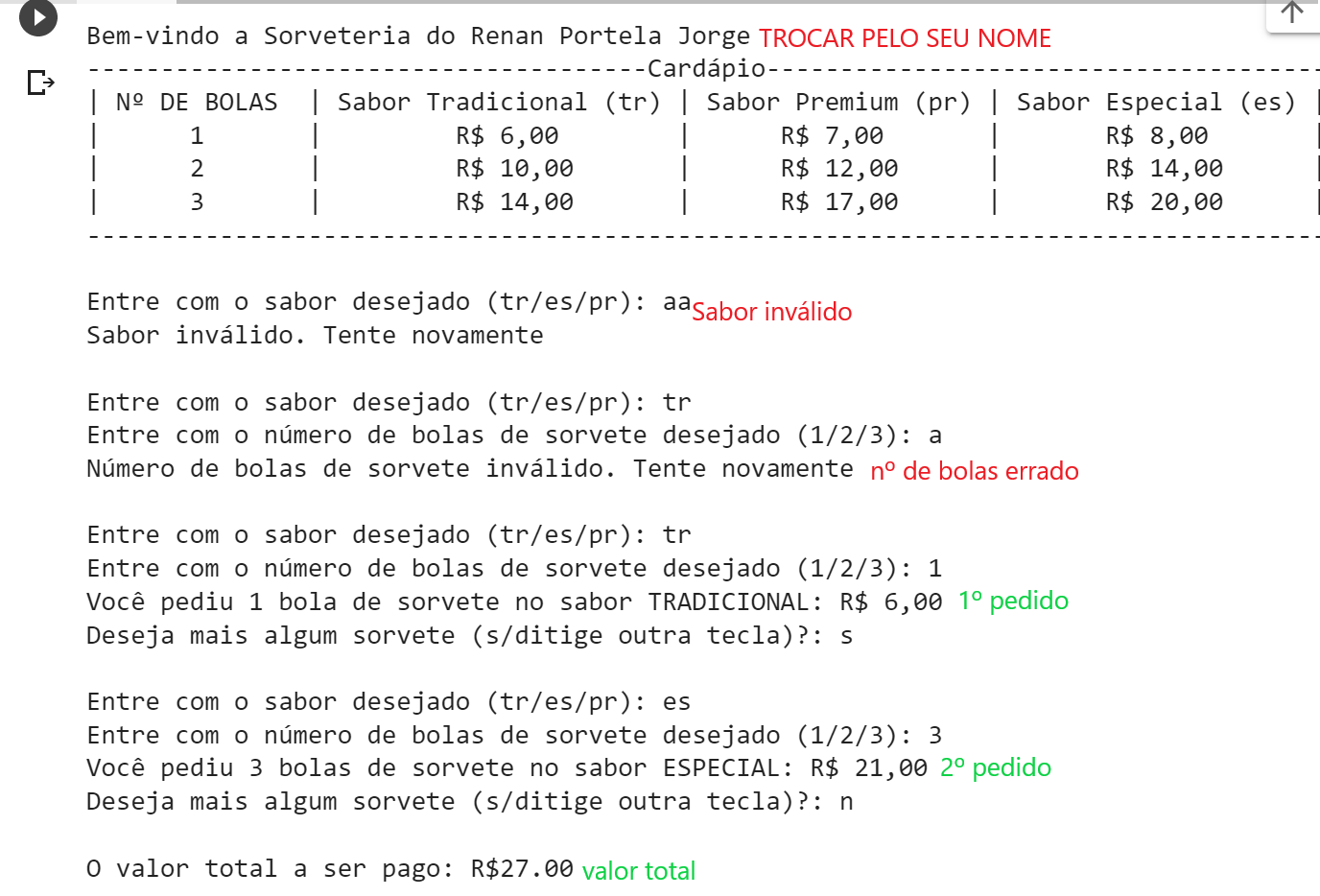
A Sorveteria possui seguinte relação:

* 1bolade sorvete no sabor tradicional (tr) custa 6 reais, no sabor premium (pr) 7 reais e no especial (es)8 reais;
* 2bolas de sorvete no sabor tradicional (tr) custam11 reais, no sabor premium (pr) 13 reais e no especial (es) 15 reais;
* 3bolas de sorvete no sabor tradicional (tr)custam15 reais, no sabor premium (pr) 18 reais e no especial (es) 21 reais;

Elabore um programa em Python que:

1. Realizar o print uma mensagem de boas-vindas que apareça o seu nome;
2. Deve-se entrar com o **sabor** (tr/pr/es) e o**número de bolas** de sorvete desejado (1/2/3)[EXIGÊNCIA DE CÓDIGO 1 de 6];
3. Deve-se executar o print da mensagem de “Quantidade de Bolas de Sorvete Inválida". Se o usuário entrar com a quantidade de bolas de sorvete diferente de 1,2 e 3**repetir a partir do item B**[EXIGÊNCIA DE CÓDIGO 2 de 6];
4. Deve-se executar o printda mensagem de“Sabor de Sorvete Inválido" se o usuário entrar com um sabor diferente de tr (tradicional), pr (premium) e es (especial). Printar: e **repetir a partir do item B**; [EXIGÊNCIA DE CÓDIGO 3 de 6];
5. Deve-se perguntar se o cliente quer pedir mais alguma coisa. Sesim **repetir a partir do item B**, senão encerrar o programa printando o **valor total**[EXIGÊNCIA DE CÓDIGO 4 de 6];
6. Deve-se utilizar as estruturas de **while, break, continue (todas elas)**[EXIGÊNCIA DE CÓDIGO5 de 6];
7. Deve-se fazer comentários no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
8. Deve-se colocar na apresentação de saída de console um pedido no qual o usuário errou ao digitar o sabor do sorvete[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];
9. Deve-se colocar na apresentação de saída de console um pedido no qual o usuário errou ao digitar o número de bolas de sorvete[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];
10. Deve-se colocar na apresentação de saída de console um pedido com duas opções sabores diferentes com quantidade de bolas diferentes [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

EXEMPLO DE SAÍDA DE CONSOLE:

s

**Figura 2: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o sabor do sorvete (tradicional,premium ou especial) a nº de bolas de sorvete (1,2ou 3). Há uma tentativa de pedido que se erro o sabor do sorvete e outra que se errou o nº de bolas de sorvete. Há também um pedido com 2 sorvetes com diferentes quantidades de bola de sorvete e sabores diferentes.**

Código da Questão 2:

nome = 'Gabriel Rodrigues de Almeida Messa'

print(f"Bem-vindo ao aplicativo de vendas da sorveteria do {nome}!")

tabela\_valores = """

Opções:

+------------------------+----------Cardápio--------+----------------+-------------+

| Número de Bolas | Sabor Tradicional | Sabor Premium | Sabor Especial |

+------------------------+----------------+----------------+----------------+------+

| 1 | R$ 6.00 | R$ 7.00 | R$ 8.00 |

| 2 | R$ 10.00 | R$ 12.00 | R$ 14.00 |

| 3 | R$ 14.00 | R$ 17.00 | R$ 20.00 |

+------------------------+----------------+----------------+----------------+------+

"""

print(tabela\_valores)

total\_pedido = 0

while True:

sabor = input("Digite o sabor do sorvete (tr/pr/es): ")

quantidade = input("Digite a quantidade de bolas de sorvete (1/2/3): ")

if quantidade not in ["1", "2", "3"]:

print("Opção de quantidade inválida")

continue

if sabor not in ["tr", "pr", "es"]:

print("Opção de sabor inválida")

continue

quantidade = int(quantidade)

if quantidade == 1:

if sabor == "tr":

preco = 6

elif sabor == "pr":

preco = 7

else:

preco = 8

elif quantidade == 2:

if sabor == "tr":

preco = 10

elif sabor == "pr":

preco = 12

else:

preco = 14

else:

if sabor == "tr":

preco = 14

elif sabor == "pr":

preco = 17

else:

preco = 20

total\_pedido += preco

print(f"Valor do pedido: R${preco:.2f}")

mais = input("Deseja pedir mais alguma coisa? (s/n): ")

if mais.lower() != "s":

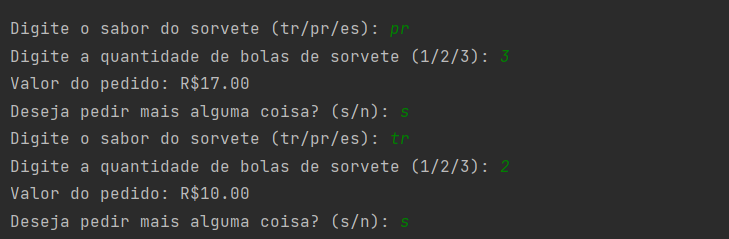
break

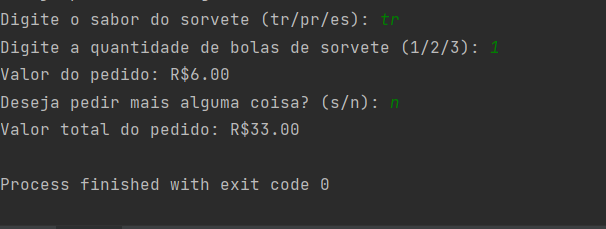
print(f"Valor total do pedido: R${total\_pedido:.2f}")

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 banho para um petshop. Você ficou com a parte de desenvolver a interface com o funcionário.

O petshop opera da seguinte maneira:

* Para cães com **peso** menor que 3 kg o valor **base** é de 40 reais;
* Para cães com **peso** igual ou maior que 3 kg e menor que 10 kg o valor **base** é de 50 reais;
* Para cães com **peso** igual ou maior que 10 kg e menor que 30kg o valor **base** é de 60 reais;
* Para cães com **peso** igual ou maior que 30 kg e menor que 50kg o valor **base** é de 70 reais;
* Para cães com **pelo** curto (c) o **multiplicador** é 1;
* Para cães com **pelo** médio (m) o **multiplicador** é 1.5;
* Para cães com **pelo** longo (l) o **multiplicador** é 2;
* Para o **adicional** de cortar unhas (1) do cachorro é cobrado um valor **extra** de 10 reais;
* Para o **adicional** de escovar os dentes (2) do cachorro é cobrado um valor **extra** de 12 reais;
* Para o **adicional** de limpar as orelhas (3) do cachorro é cobrado um valor **extra** de 15 reais;
* Para o **adicional** de não querer mais nada (0) é cobrado um valor **extra** de 0 reais;

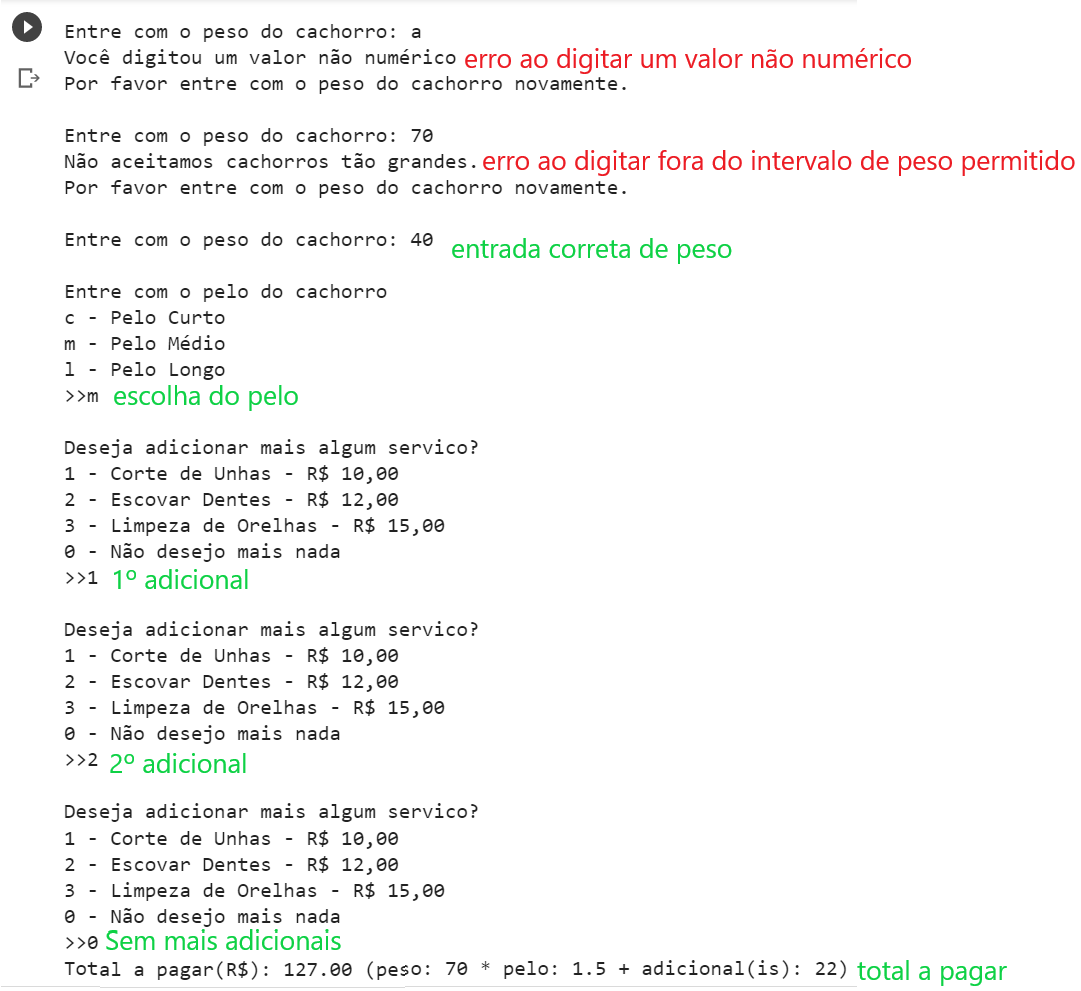
O valor final da conta é calculado da seguinte maneira:

total = **base \* multiplicador + extra**

Elabore um programa em Python que:

1. Realizar o print uma mensagem de boas-vindas que apareça o seu nome;
2. Deve-se criar uma função chamada **cachorro\_peso()**em que**:**[EXIGÊNCIA DE CÓDIGO 1 de 6];
   1. Pergunta o **peso** do cachorro;
   2. Retorna o valor **base** com base no peso;
   3. Repete a pergunta do item B.a se peso for igual ou acima 50kg;
   4. Repete a pergunta do item B.a se digitar um valor não numérico;
3. Deve-se criar uma função chamada **cachorro\_pelo()** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 6];
   1. Pergunta o **pelo** do cachorro;
   2. Retorna o **multiplicador** com base nos itens descritos no enunciado;
   3. Repete a pergunta do item C.a se digitar uma opção diferente de: c/m/l;
4. Deve-se criar uma função chamada **cachorro\_extra()** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 6];
   1. Pergunta pelo serviço **adicional**;
   2. Acumular o valor **extra** de cada **adicional;**
   3. Repetir a pergunta item D enquanto não se digitar opção de: "não querer mais nada (0)";
   4. Quando digitar o **adicional** não querer mais nada (0) retornar o valor **extra;**
5. Deve-se calcular o total a pagar na parte do mainconforme descrito no enunciado[EXIGÊNCIA DE CÓDIGO 4 de 6];
6. Deve-se utilizar **try/except** [EXIGÊNCIA DE CÓDIGO 5 de 6];
7. Deve-se fazer comentários no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
8. Deve-se colocar na apresentação de saída de console um pedido no qual o usuário digitou um valor não numérico para o peso [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];
9. Deve-se colocar na apresentação de console um pedido no qual o usuário digitou um valor acima 50 para o peso [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];
10. Deve-se colocar na apresentação de console um pedido no qual o peso e o tipo de pelo sejam válidos e com mais 2 extras [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

EXEMPLO DE SAÍDA DE CONSOLE:



**Figura 3: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o peso do cachorro o tipo de pelo (c, m ou l). Há um pedido que inicialmente o usuário digitou um valor não numérico (a) e depois um valor acima de 50 (70). Na sequência, o usuário digitou um valor de peso válido, de tipo de peso válido e 2 adicionais.**

Código da Questão 3:

nome = "Gabriel Rodrigues de Almeida Messa"

print(f"Bem-vindo, {nome}!")

def entrada\_peso():

print("Escolha o peso do cachorro:")

print("Menor que 3 kg - Valor base: R$40")

print("3 kg ou mais, e menor que 10 kg - Valor base: R$50")

print("10 kg ou mais, e menor que 30 kg - Valor base: R$60")

print("30 kg ou mais, e menor que 50 kg - Valor base: R$70")

while True:

try:

peso = float(input("Digite o peso do cachorro em kg: "))

if peso < 3:

return 40

elif peso < 10:

return 50

elif peso < 30:

return 60

elif peso < 50:

return 70

else:

print("Peso muito alto. Digite novamente.")

except ValueError:

print("Valor inválido. Digite novamente.")

def entrada\_pelo():

print("Escolha o tipo de pelo:")

print("c - Pelo curto (Multiplicador: 1)")

print("m - Pelo médio (Multiplicador: 1.5)")

print("l - Pelo longo (Multiplicador: 2)")

while True:

pelo = input("Digite o tipo de pelo do cachorro (c/m/l): ")

if pelo == "c":

return 1

elif pelo == "m":

return 1.5

elif pelo == "l":

return 2

else:

print("Tipo de pelo inválido. Digite novamente.")

def entrada\_adicionais():

print("Escolha o serviço adicional:")

print("1 - Cortar unhas (R$10)")

print("2 - Escovar dentes (R$12)")

print("3 - Limpar orelhas (R$15)")

print("0 - Não querer mais nada (R$0)")

extra = 0

for \_ in range(1):

while True:

try:

adicional = int(input("Digite o número do serviço adicional: "))

if adicional == 1:

extra += 10

print("Cortar unhas")

elif adicional == 2:

extra += 12

print("Escovar dentes")

elif adicional == 3:

extra += 15

print("Limpar orelhas")

elif adicional == 0:

print("Não querer mais nada")

break

except ValueError:

print("Valor inválido. Digite novamente.")

return extra

# Solicitar informações

peso = entrada\_peso()

pelo = entrada\_pelo()

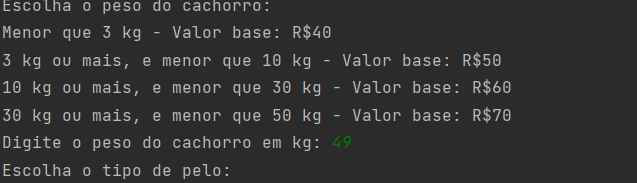
extra = entrada\_adicionais()

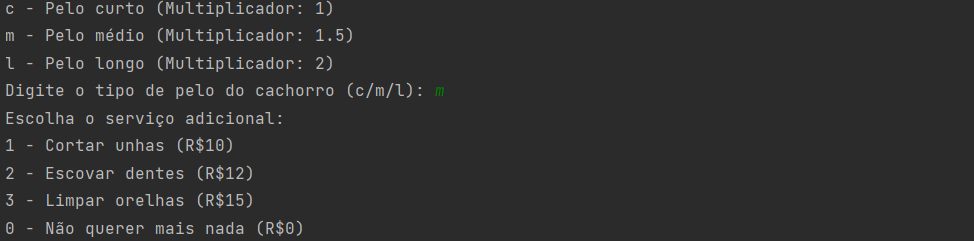
# Calcular total e mostrar resultado

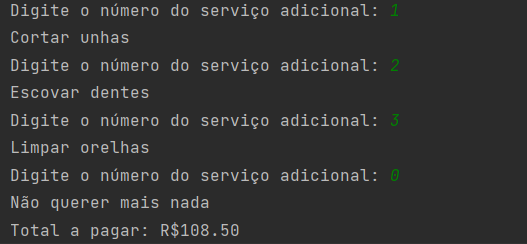
total = extra + pelo + peso

print(f"Total a pagar: R${total:.2f}")

Saída do Console da Questão 3:







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

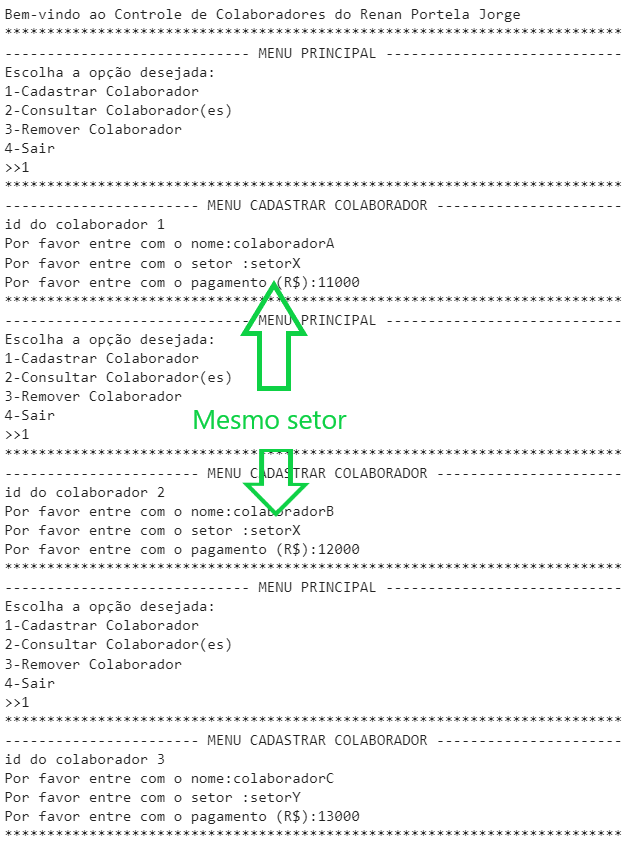
Enunciado: Você e sua equipe de programadores foram contratados por pequena empresa para desenvolver o software de gerencial de pessoas. Este software deve ter o seguinte menu e opções:

1. Cadastrar Colaborador
2. Consultar Colaborador
   1. Consultar Todos
   2. Consultar por Id;
   3. Consultar por Setor;
   4. Retornar ao menu;
3. Remover Colaborador
4. Encerrar Programa

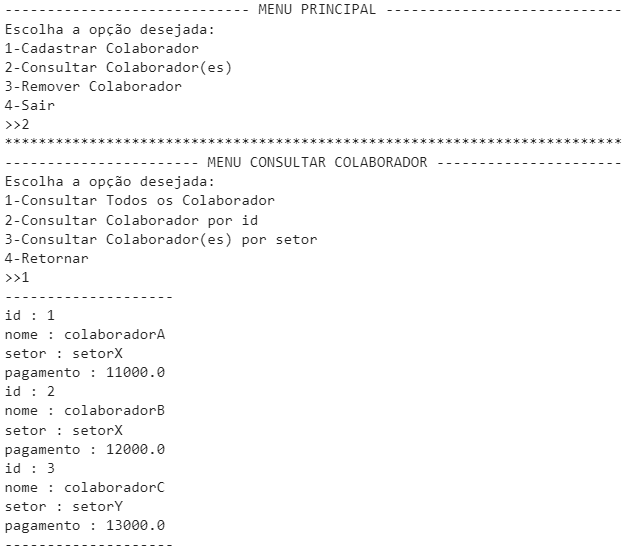
Elabore um programa em Python que:

1. Realizar o print uma mensagem de boas-vindas que apareça o seu nome;
2. Deve-se criar uma lista vazia com o nome de **lista\_colaboradores** e a variável **id\_global** com valor inicial igual a 0 [EXIGÊNCIA DE CÓDIGO 1 de 7];
3. Deve-se criar uma função chamada **cadastrar\_colaborador(id)** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Pergunta **nome**,**setor**, **pagamento** do colaborador;
   2. Armazena o**id** (este é fornecido via parâmetro da função),**nome**, **setor**, **salário** dentro de um dicionário;
   3. Copiar o dicionário dentro para dentro da **lista\_colaboradores**;
4. Deve-se criar uma função chamada **consultar\_colaborador()** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Deve-se pergunta qual opção deseja (1. Consultar Todos /2. Consultar por Id/ 3. Consultar por Setor / 4. Retornar ao menu) e realizar o print “Opção inválida" se entrar com valor diferente de 1, 2, 3 ou 4:
      1. Se Consultar Todos, apresentar todos os colaboradores com todos os seus dados cadastrados;
      2. Se Consultar por Id, apresentar ocolaborador específico com todos os seus dados cadastrados;
      3. Se Consultar por Setor, apresentar todos os colaboradores do setor específico com todos os seus dados cadastrados;
      4. Se Retornar ao menu, deve-se retornar ao menu principal
5. Deve-se criar uma função chamada **remover\_colaborador()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Deve-se pergunta pelo **id** do colaborador a ser removido;
   2. Remover o colaborador da **lista\_colaboradores**;
6. Deve-se criar uma estrutura de menu no main em que: [EXIGÊNCIA DE CÓDIGO 5 de 7];
   1. Deve-se pergunta qual opção deseja (1. Cadastrar Colaborador / 2. Consultar Colaborador / 3. Remover Colaborador / 4. Encerrar Programa)e realizar o print “Opção inválida" se entrar com valor diferente de 1, 2, 3 ou 4:
      1. Se Cadastrar Colaborador, acrescentar em uma variavel **id\_ global** e chamar a função **cadastrar\_colaborador(id\_ global)**;
      2. Se Consultar Colaborador, chamar função **consultar\_colaborador()**;
      3. Se Remover Colaborador, chamar função **remover\_colaborador()**;
      4. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
7. Deve-se utilizar **lista de dicionários**(uma lista contento dicionários dentro)[EXIGÊNCIA DE CÓDIGO 6 de 7];
8. Deve-se fazer comentários no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
9. Deve-se colocar na apresentação de saída de console o cadastro de 3 colaboradores (sendo **2** deles no mesmo setor) [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
10. Deve-se colocar na apresentação de saída de console a consulta de todos os colaboradores [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de ];
11. Deve-se colocar na apresentação de saída de console a consulta por código de um dos colaboradores[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
12. Deve-se colocar na apresentação de saída de console a consulta por setor em que **2** colaboradores façam parte [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
13. Deve-se colocar na apresentação de saída de console a remoção de um dos colaboradores e na sequência a consulta de todos os colaboradores[EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

EXEMPLO DE SAÍDA DE CONSOLE:



**Figura 4.1: Exemplo de saída de console que o aluno deve fazer. Em que se cadastra 3 colaboradores, sendo 2 num setorX e outro no setorY.**



**Figura 4.2: Exemplo de saída de console que o aluno deve fazer. Em que se consulta todos os colaboradores.**

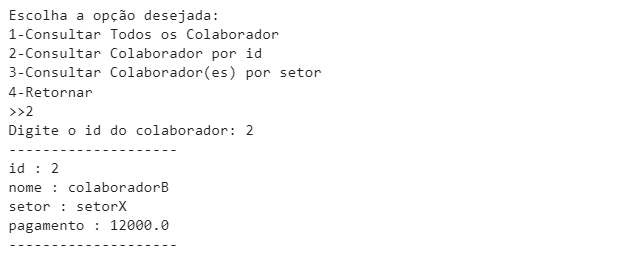


Figura 4.3: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o colaborador de id 2.

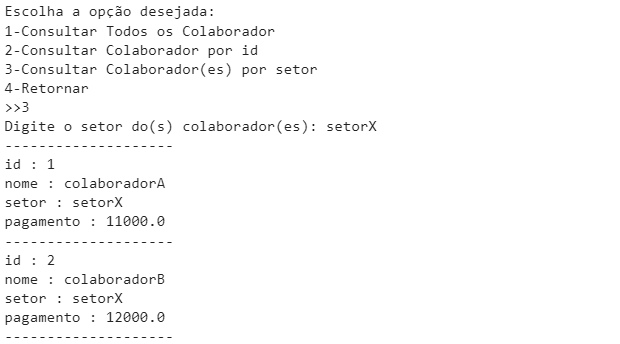


Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta os colaboradores do setor setorX.

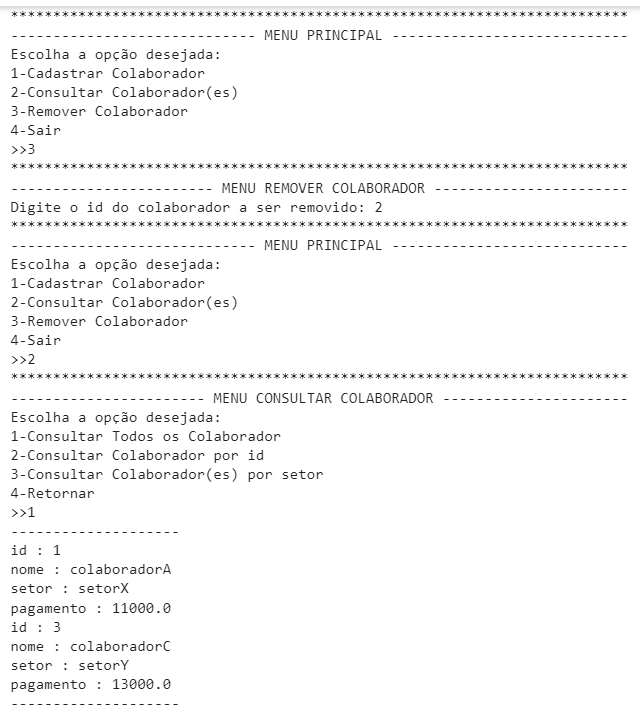


Figura 4.5: Exemplo de saída de console que o aluno deve fazer. Em que se removeu o colaborador de id 2 e na sequência, consultou-se todos os colaboradores.

Código da Questão 4:

# Imprime uma mensagem de boas-vindas ao iniciar o programa.  
Seu\_nome = 'Gabriel Rodrigues de Almeida Messa'  
print(f"Bem-vindo ao software de gerenciamento de pessoas por {Seu\_nome}!")  
  
# Inicializa uma lista vazia para armazenar os colaboradores e uma variável id\_global com valor inicial zero.  
# A lista armazena os dados dos colaboradores e id\_global é usado para atribuir IDs exclusivos aos colaboradores.  
lista\_colaboradores = []  
id\_global = 0  
  
# Função para cadastrar colaboradores. Ela aceita um ID como parâmetro, que é o valor atual de id\_global.  
def cadastrar\_colaborador(id\_global):  
 # Coleta informações básicas do colaborador.  
 print("\nCadastro de Novo Colaborador:")  
 nome = input("Digite o nome completo do colaborador: ")  
 setor = input("Digite o setor em que o colaborador trabalha: ")  
  
 # Validação do valor de pagamento para garantir que seja um número.  
 while True:  
 try:  
 pagamento = float(input("Digite o pagamento mensal do colaborador: "))  
 break  
 except ValueError:  
 print("Valor inválido. Digite um número válido para o pagamento.")  
  
 # Cria um dicionário com as informações do colaborador e adiciona à lista de colaboradores.  
 colaborador = {  
 "id": id\_global,  
 "nome": nome,  
 "setor": setor,  
 "pagamento": pagamento  
 }  
 lista\_colaboradores.append(colaborador)  
 print("Colaborador cadastrado com sucesso!")  
  
# Função para consultar colaboradores. Pode buscar todos os colaboradores, por ID específico ou por setor.  
def consultar\_colaborador():  
 while True:  
 # Apresenta as opções de consulta ao usuário.  
 print("\nOpções de Consulta:")  
 print("1. Consultar Todos")  
 print("2. Consultar por Id")  
 print("3. Consultar por Setor")  
 print("4. Retornar ao menu")  
  
 # Captura a escolha do usuário e realiza a ação correspondente.  
 try:  
 opcao\_consulta = int(input("Escolha uma opção de consulta: "))  
 if opcao\_consulta in [1, 2, 3, 4]:  
 if opcao\_consulta == 1:  
 print("\nListagem de Todos os Colaboradores:")  
 print("ID | Nome | Setor | Pagamento")  
 for colaborador in lista\_colaboradores:  
 print(f"{colaborador['id']} | {colaborador['nome']} | {colaborador['setor']} | {colaborador['pagamento']}")  
 elif opcao\_consulta == 2:  
 print("\nConsulta por ID:")  
 id\_busca = int(input("Digite o id do colaborador: "))  
 # Flag para verificar se o colaborador foi encontrado.  
 encontrado = False  
 for colaborador in lista\_colaboradores:  
 if colaborador['id'] == id\_busca:  
 print(f"ID: {colaborador['id']}, Nome: {colaborador['nome']}, Setor: {colaborador['setor']}, Pagamento: {colaborador['pagamento']}")  
 encontrado = True  
 break  
 if not encontrado:  
 print("Colaborador não encontrado.")  
 elif opcao\_consulta == 3:  
 print("\nConsulta por Setor:")  
 setor\_busca = input("Digite o setor: ")  
 # Filtra colaboradores pelo setor e exibe os encontrados.  
 colaboradores\_setor = [col for col in lista\_colaboradores if col['setor'] == setor\_busca]  
 if colaboradores\_setor:  
 print(f"Colaboradores do setor {setor\_busca}:")  
 for col in colaboradores\_setor:  
 print(f"{col['id']} | {col['nome']} | {col['setor']} | {col['pagamento']}")  
 else:  
 print(f"Nenhum colaborador encontrado no setor '{setor\_busca}'.")  
 elif opcao\_consulta == 4:  
 # Retorna ao menu principal.  
 return  
 else:  
 print("Opção inválida.")  
  
 except ValueError:  
 print("Entrada inválida. Por favor, digite um número válido para a opção desejada.")  
  
# Função para remover colaboradores.  
def remover\_colaborador():  
 print("\nRemoção de Colaborador:")  
 id\_remover = int(input("Digite o ID do colaborador a ser removido: "))  
 # Flag para verificar se o colaborador foi removido.  
 removido = False  
 for colaborador in lista\_colaboradores:  
 if colaborador['id'] == id\_remover:  
 lista\_colaboradores.remove(colaborador)  
 removido = True  
 print("Colaborador removido com sucesso!")  
 break  
 if not removido:  
 print("Colaborador não encontrado.")  
  
# Loop principal para execução do programa.  
while True:  
 # Apresenta o menu principal e as opções disponíveis.  
 print("\nMenu Principal:")  
 print("1. Cadastrar Colaborador")  
 print("2. Consultar Colaborador")  
 print("3. Remover Colaborador")  
 print("4. Encerrar Programa")  
  
 # Captura a escolha do usuário e realiza a ação correspondente.  
 try:  
 opcao\_menu = int(input("Escolha uma opção: "))  
 if opcao\_menu in [1, 2, 3, 4]:  
 if opcao\_menu == 1:  
 # Incrementa o ID global para cada novo colaborador e chama a função de cadastro.  
 id\_global += 1  
 cadastrar\_colaborador(id\_global)  
 elif opcao\_menu == 2:  
 # Chama a função de consulta.  
 consultar\_colaborador()  
 elif opcao\_menu == 3:  
 # Chama a função de remoção.  
 remover\_colaborador()  
 elif opcao\_menu == 4:  
 # Encerra o programa.  
 print("Encerrando o programa.")  
 break  
 else:  
 print("Opção inválida.")  
  
 except ValueError:  
 print("Entrada inválida. Por favor, digite um número válido para a opção desejada.")

Saída do Console da Questão 4:

