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

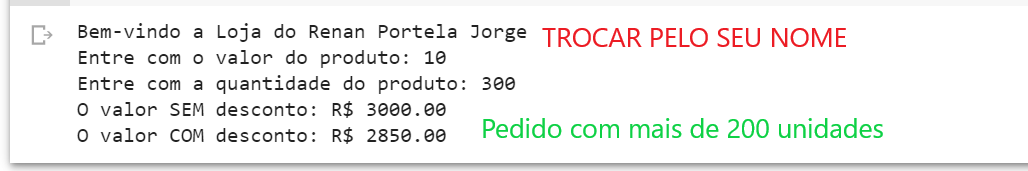
Enunciado: Imagina-se que você é um dos programadores responsáveis pela construção de app de vendas para uma determinada empresa X que vende em atacado. Uma das estratégias de vendas dessa empresa X é dar desconto maiores por unidade as informações abaixo:

* 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 que 1000 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 console um pedido recebendo desconto [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 1];

EXEMPLO DE SAÍDA DE CONSOLE:

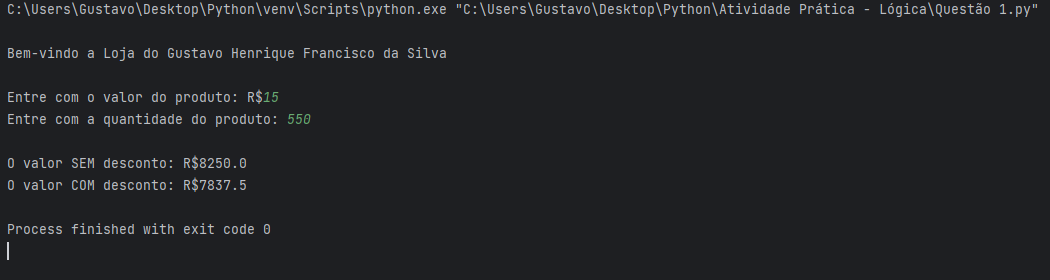


**Figura 1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o valor do produto (pode ser qualquer valor) a quantidade (deve ser maior que 200) e apresenta o valor final sem o desconto e com o desconto.**

Código da Questão 1:

print('\nBem-vindo a Loja do Gustavo Henrique Francisco da Silva')  
  
# Solicita o preço do produto e sua quantidade e atribui as suas respectivas variáveis.  
preco = float(input('\nEntre com o valor do produto: R$'))  
quant = int(input('Entre com a quantidade do produto: '))  
  
# Multiplicando o preço que foi inserido pela quantidade do produto para achar o valor total da compra.  
total = preco \* quant  
  
if preco > 0 and quant > 0:  
  
 # O método usado para aplicar o desconto foi utilizando a fórmula: [valor total \* (desconto / 100)] para achar  
 # o valor à ser descontado e em seguida subtrair esse resultado do valor total original.  
 if quant < 200:  
 print('\nO valor SEM desconto: R${}'.format(total))  
 print('O valor COM desconto: R${}'.format(total))  
 elif 200 <= quant < 1000:  
 print('\nO valor SEM desconto: R${}'.format(total))  
 print('O valor COM desconto: R${}'.format(total - (total \* (5 / 100))))  
 elif 1000 <= quant < 2000:  
 print('\nO valor SEM desconto: R${}'.format(total))  
 print('O valor COM desconto: R${}'.format(total - (total \* (10 / 100))))  
 elif 2000 <= quant:  
 print('\nO valor SEM desconto: R${}'.format(total))  
 print('O valor COM desconto: R${}'.format(total - (total \* (15 / 100))))

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.

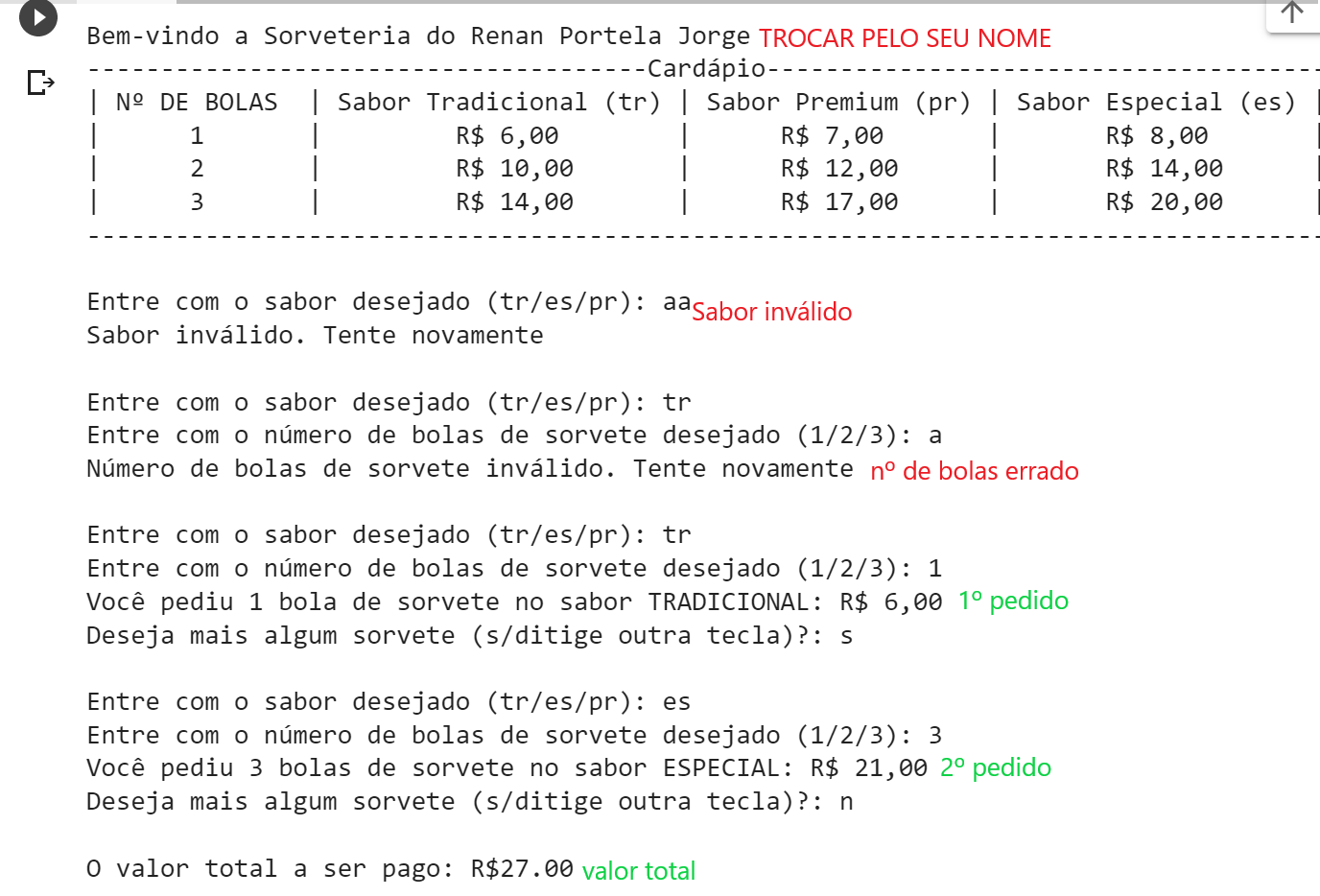
A Sorveteria possui seguinte relação:

* 1 bola de sorvete no sabor tradicional (tr) custa 6 reais, no sabor premium (pr) 7 reais e no especial (es) 8 reais;
* 2 bolas de sorvete no sabor tradicional (tr) custam 11 reais, no sabor premium (pr) 13 reais e no especial (es) 15 reais;
* 3 bolas de sorvete no sabor tradicional (tr) custam 15 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 print da 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. Se sim **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 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 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:

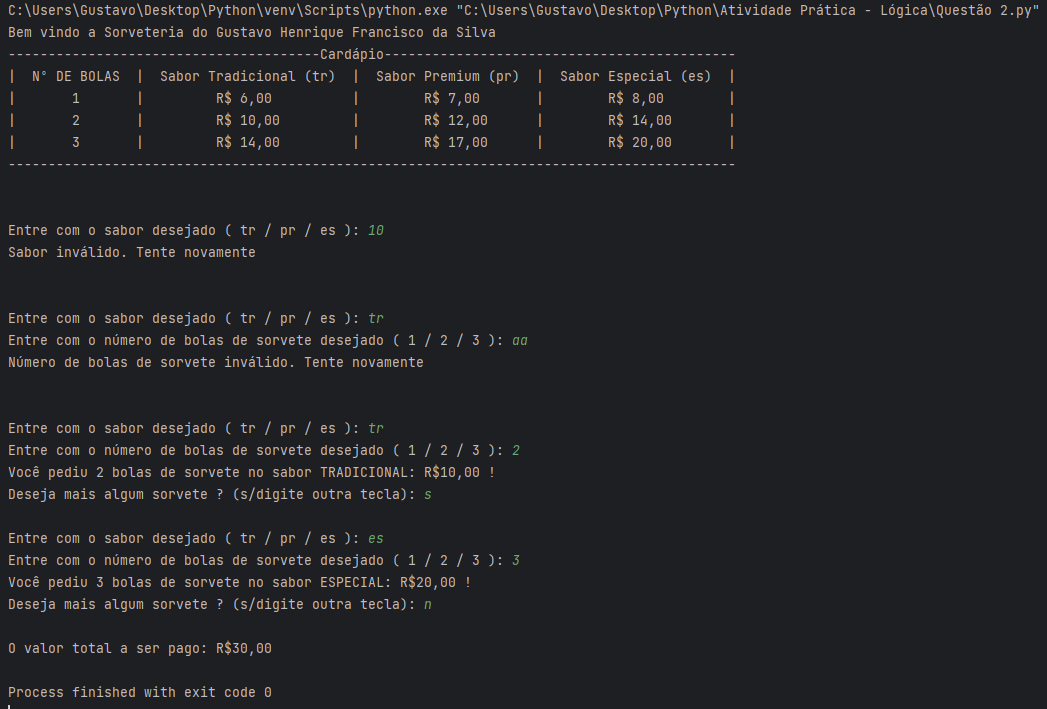


**Figura 2: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o sabor do sorvete (tradicional, premium ou especial) a nº de bolas de sorvete (1,2 ou 3). Há uma tentativa de pedido que se erro o sabor do sorvete e outra que se errou o nº de bolas de sorvete. Há também um pedido com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

Código da Questão 2:

# Cardápio da Sorveteria  
print('Bem vindo a Sorveteria do Gustavo Henrique Francisco da Silva')  
print('---------------------------------------Cardápio--------------------------------------------')  
print('| Nº DE BOLAS | Sabor Tradicional (tr) | Sabor Premium (pr) | Sabor Especial (es) |')  
print('| 1 | R$ 6,00 | R$ 7,00 | R$ 8,00 |')  
print('| 2 | R$ 10,00 | R$ 12,00 | R$ 14,00 |')  
print('| 3 | R$ 14,00 | R$ 17,00 | R$ 20,00 |')  
print('-------------------------------------------------------------------------------------------\n')  
  
valor = 0  
total = 0  
  
# Onde é feito o pedido especificando a quantidade e o sabor em um loop que se quebra quando o que foi inserido  
# é válido ou continua a perguntar quanto o dado é inválido  
while True:  
 while True:  
 sabor = input('\nEntre com o sabor desejado ( tr / pr / es ): ')  
 if sabor == 'tr' or sabor == 'pr' or sabor == 'es':  
 bolas = input('Entre com o número de bolas de sorvete desejado ( 1 / 2 / 3 ): ')  
 if bolas == '1' or bolas == '2' or bolas == '3':  
 break  
 else:  
 print('Número de bolas de sorvete inválido. Tente novamente \n')  
 continue  
 else:  
 print('Sabor inválido. Tente novamente\n')  
 continue  
  
# Na parte a seguir foi definido o valor de cada sabor de sorvete de acordo com a quantidade bolas  
 if sabor == 'tr':  
 if bolas == '1':  
 valor = 6  
 elif bolas == '2':  
 valor = 10  
 elif bolas == '3':  
 valor = 14  
  
 elif sabor == 'pr':  
 if bolas == '1':  
 valor = 7  
 elif bolas == '2':  
 valor = 12  
 else:  
 valor = 17  
  
 elif sabor == 'es':  
 if bolas == '1':  
 valor = 8  
 elif bolas == '2':  
 valor = 14  
 else:  
 valor = 20  
  
 if sabor == 'tr':  
 print('Você pediu {} bolas de sorvete no sabor TRADICIONAL: R${},00 !'.format(bolas, valor))  
 elif sabor == 'pr':  
 print('Você pediu {} bolas de sorvete no sabor PREMIUM: R${},00 !'.format(bolas, valor))  
 elif sabor == 'es':  
 print('Você pediu {} bolas de sorvete no sabor ESPECIAL: R${},00 !'.format(bolas, valor))  
  
# Aqui é feito o calculo do valor total das compras, em que a cada novo pedido de sorvete o valor é adicionado a variável "total"  
 total += valor  
  
 novacompra = input('Deseja mais algum sorvete ? (s/digite outra tecla): ')  
 if novacompra == 's':  
 continue  
 else:  
 break  
  
#Mostrado na tela o valor total do pedido  
print('\nO valor total a ser pago: R${},00'.format(total))

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;

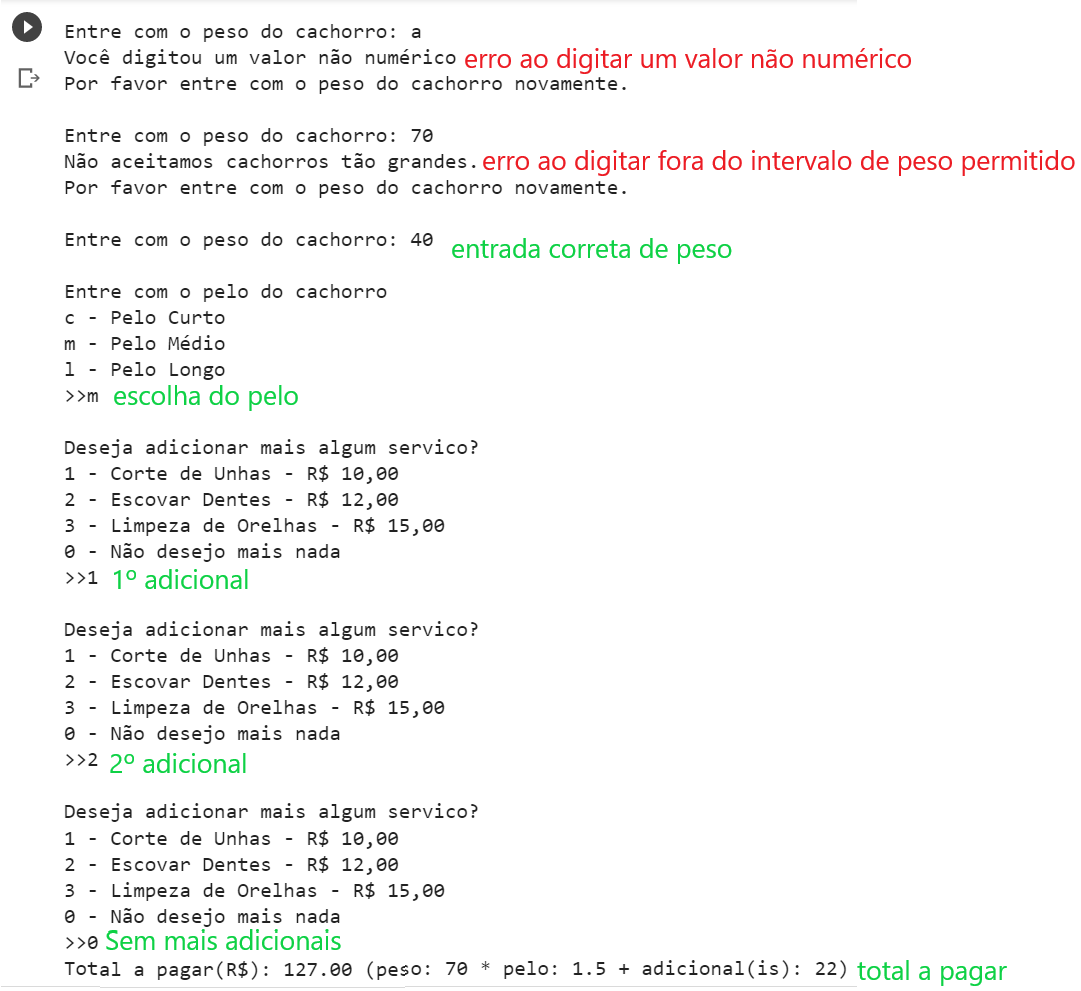
O valor final da conta é calculado da seguinte maneira:

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 ovalor **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.a 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 main conforme 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:

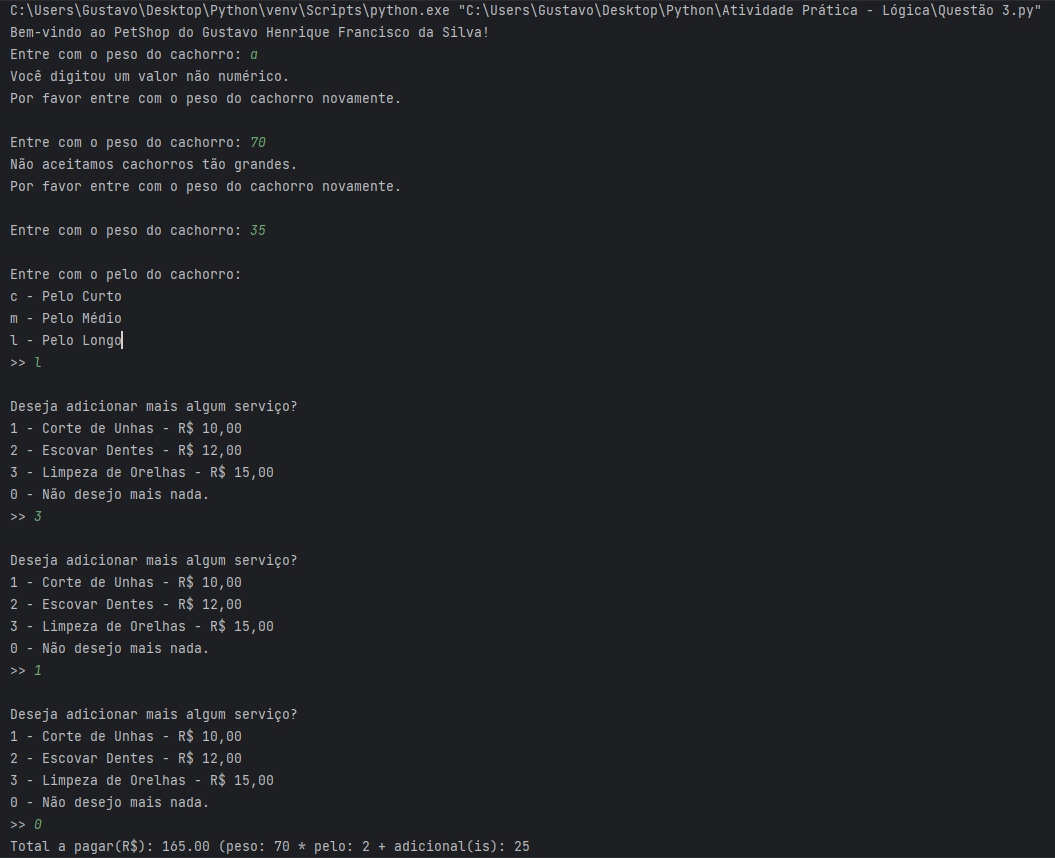


**Figura 3: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o peso do cachorro o tipo de pelo (c, m ou l). Há um pedido que inicialmente o usuário digitou um valor não numérico (a) e depois um valor acima de 50 (70). Na sequência, o usuário digitou um valor de peso válido, de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

Código da Questão 3:

print("Bem-vindo ao PetShop do Gustavo Henrique Francisco da Silva!")  
  
# Função para calcular o valor base de acordo com o peso do cachorro:  
def cachorro\_peso():  
 while True:  
 try:  
 peso = float(input("Entre com o peso do cachorro: "))  
 if peso < 3:  
 return 40  
 elif 3 <= peso < 10:  
 return 50  
 elif 10 <= peso < 30:  
 return 60  
 elif 30 <= peso < 50:  
 return 70  
 else:  
 print("Não aceitamos cachorros tão grandes. \nPor favor entre com o peso do cachorro novamente.\n")  
 except ValueError:  
 print("Você digitou um valor não numérico. \nPor favor entre com o peso do cachorro novamente.\n")  
  
  
# Função para definir o multiplicador de acordo com o tipo de pelo do cachorro:  
def cachorro\_pelo():  
 while True:  
 pelo = input("\nEntre com o pelo do cachorro:\nc - Pelo Curto\nm - Pelo Médio\nl - Pelo Longo\n>> ")  
 if pelo == 'c':  
 return 1  
 elif pelo == 'm':  
 return 1.5  
 elif pelo == 'l':  
 return 2  
 else:  
 print("Opção inválida. Por favor, digite 'c' para pelo curto, 'm' para pelo médio ou 'l' para pelo longo.")  
  
  
# Função para calcular o valor extra com base nos serviços adicionais  
def cachorro\_extra():  
 valor\_extra = 0  
 while True:  
 try:  
 adicional = int(input("\nDeseja adicionar mais algum serviço? \n1 - Corte de Unhas - R$ 10,00 \n2 - Escovar Dentes - R$ 12,00 \n3 - Limpeza de Orelhas - R$ 15,00 \n0 - Não desejo mais nada. \n>> "))  
 if adicional == 0:  
 return valor\_extra  
 elif adicional == 1:  
 valor\_extra += 10  
 elif adicional == 2:  
 valor\_extra += 12  
 elif adicional == 3:  
 valor\_extra += 15  
 else:  
 print("Opção inválida. Por favor, escolha um serviço adicional válido.")  
 except ValueError:  
 print("Valor inválido. Por favor, escolha um serviço adicional válido.")  
  
# Um "Try" onde é feito o cálculo do valor total usando o peso, o tipo de pelo e os serviços adicionais e em seguida  
# retorna o total a pagar na tela.  
try:  
 base = cachorro\_peso()  
 multiplicador = cachorro\_pelo()  
 valor\_extra = cachorro\_extra()  
  
 total = base \* multiplicador + valor\_extra  
 print(f"Total a pagar(R$): {total:.2f} (peso: {base} \* pelo: {multiplicador} + adicional(is): {valor\_extra}")  
except:  
 print("Ocorreu um erro inesperado.")

Saída do Console da Questão 3:



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

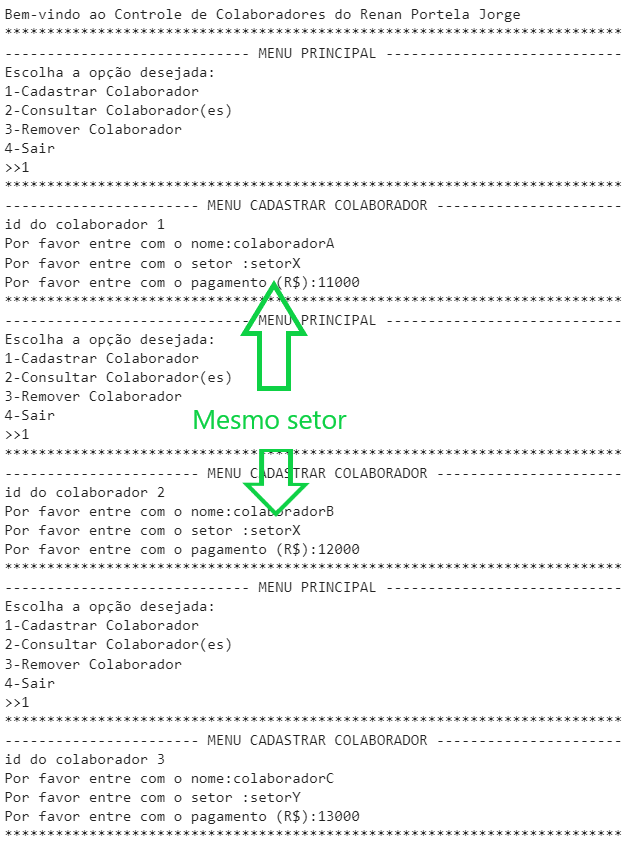
Enunciado: Você e sua equipe de programadores foram contratados por pequena empresa para desenvolver o software de gerencialme de pessoas. Este software deve ter o seguinte menu e opções:

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

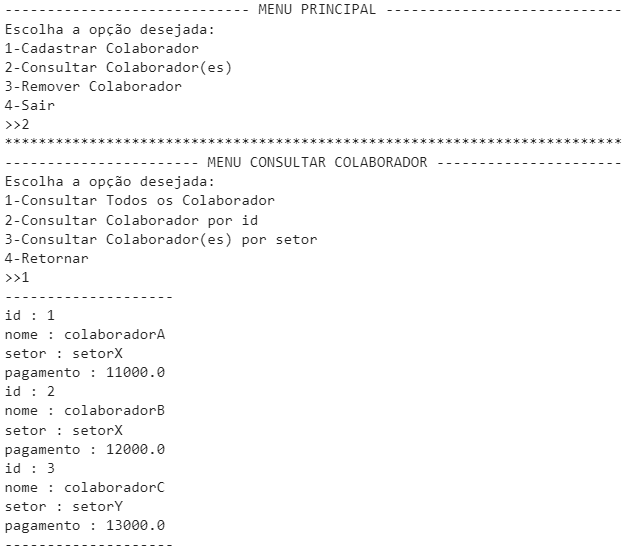
Elabore um programa em Python que:

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 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 o colaborador 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 um a 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.**

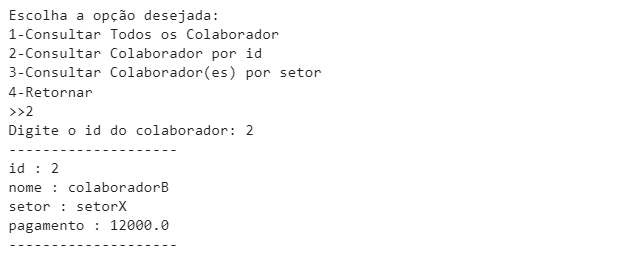


Figura 4.3: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o colaborador de id 2.

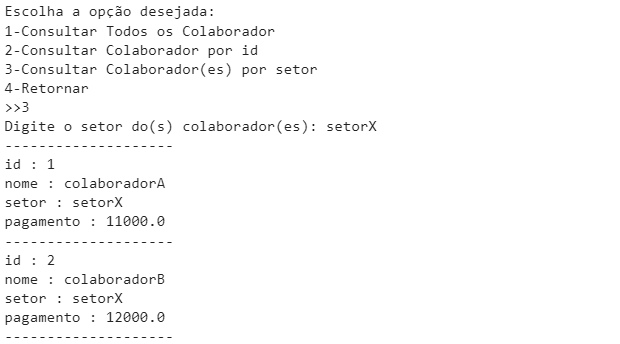


Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta os colaboradores do setor setorX.

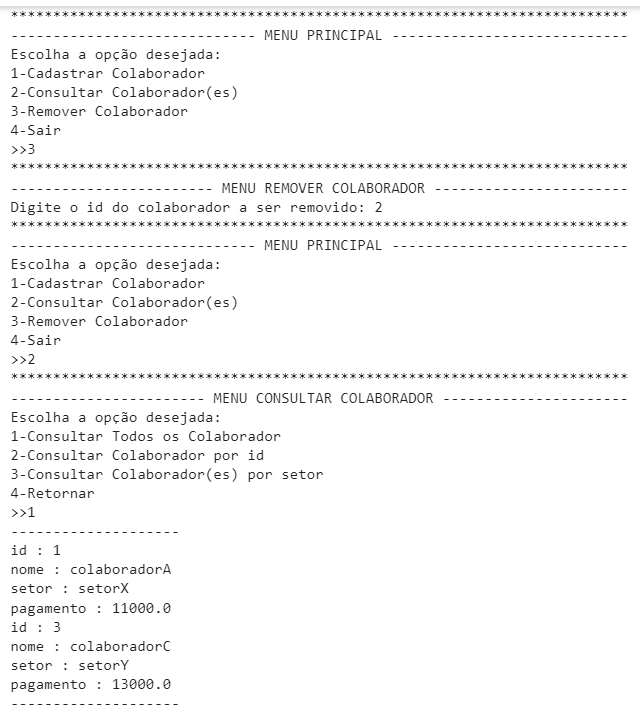


Figura 4.5: Exemplo de saída de console que o aluno deve fazer. Em que se removeu o colaborador de id 2 e na sequência, consultou-se todos os colaboradores.

Código da Questão 4:

# Função para cadastrar um novo colaborador  
def cadastrar\_colaborador(id):  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("-------------------------MENU CADASTRAR COLABORADOR---------------------------")  
 print(f"id do colaborador {id\_global}")  
 nome = input("Por favor entre com o nome: ")  
 setor = input("Por favor entre com o setor: ")  
 pagamento = float(input("Por favor entre com o pagamento (R$): "))  
  
 colaborador = {"ID": id, "Nome": nome, "Setor": setor, "Pagamento": pagamento}  
 lista\_colaboradores.append(colaborador)  
 print("Colaborador cadastrado com sucesso!")  
  
  
# Função para consultar colaboradores  
def consultar\_colaborador():  
 while True:  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("-------------------------MENU CONSULTAR COLABORADOR---------------------------")  
 print("Opções de consulta:")  
 print("1 - Consultar Todos os Colaboradores")  
 print("2 - Consultar Colaborador por ID")  
 print("3 - Consultar Colaborador(es) por setor")  
 print("4 - Retornar.")  
  
 opcao = input(">> ")  
  
 if opcao == "1":  
 for colaborador in lista\_colaboradores:  
 print("---------------------------------------")  
 print(f"ID: {colaborador['ID']}\nNome: {colaborador['Nome']}\nSetor: {colaborador['Setor']}\nPagamento: R$ {colaborador['Pagamento']:.2f}")  
 print("---------------------------------------")  
 elif opcao == "2":  
 id\_colaborador = int(input("Digite o ID do colaborador: "))  
 encontrado = False  
 for colaborador in lista\_colaboradores:  
 if colaborador["ID"] == id\_colaborador:  
 print("---------------------------------------")  
 print(f"ID: {colaborador['ID']}\nNome: {colaborador['Nome']}\nSetor: {colaborador['Setor']}\nPagamento: R$ {colaborador['Pagamento']:.2f}")  
 print("---------------------------------------")  
 encontrado = True  
 break  
 if not encontrado:  
 print("Colaborador não encontrado.")  
 elif opcao == "3":  
 setor = input("Digite o setor do(s) colaborador(es): ")  
 for colaborador in lista\_colaboradores:  
 if colaborador["Setor"] == setor:  
 print("---------------------------------------")  
 print(f"ID: {colaborador['ID']}\nNome: {colaborador['Nome']}\nSetor: {colaborador['Setor']}\nPagamento: R$ {colaborador['Pagamento']:.2f}")  
 print("---------------------------------------")  
 elif opcao == "4":  
 break  
 else:  
 print("Opção inválida.")  
  
  
# Função para remover um colaborador  
def remover\_colaborador():  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("--------------------------MENU REMOVER COLABORADOR----------------------------")  
 id\_colaborador = int(input("Digite o ID do colaborador a ser removido: "))  
 for colaborador in lista\_colaboradores:  
 if colaborador["ID"] == id\_colaborador:  
 lista\_colaboradores.remove(colaborador)  
 print("Colaborador removido com sucesso!")  
 return  
 print("Colaborador não encontrado.")  
  
  
# Variáveis globais  
lista\_colaboradores = []  
id\_global = 1  
  
#Menu Principal com cada opção desponível  
print("Bem-vindo ao Controle de Colaboradores do Gustavo Henrique Francisco da Silva!")  
while True:  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("-------------------------------MENU PRINCIPAL---------------------------------")  
 print("1 - Cadastrar Colaborador")  
 print("2 - Consultar Colaborador(es)")  
 print("3 - Remover Colaborador")  
 print("4 - Sair.")  
 opcao = input(">> ")  
  
 if opcao == "1":  
 cadastrar\_colaborador(id\_global)  
 id\_global += 1  
 elif opcao == "2":  
 consultar\_colaborador()  
 elif opcao == "3":  
 remover\_colaborador()  
 elif opcao == "4":  
 print("Encerrando o programa.")  
 break  
 else:  
 print("Opção inválida. Tente novamente.")

Saída do Console da Questão 4:

