



| **QUESTÃO 1 de 4 (25 pontos)**  **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 conforme a tabela abaixo:**   | Quantidades | Desconto | | --- | --- | | Até 9 | **0% na unidade** | | Entre 10 e 99 | **5% na unidade** | | Entre 100 e 999 | **10% na unidade** | | De 1000 para mais | **15% na unidade** |   **Elabore um programa em Python que:**   1. **Entre com o valor unitário do produto (Lembrar que número decimal é feito com ponto e não vírgula);** 2. **Entre com a quantidade desse produto;** 3. **O programa deve retornar o valor total sem desconto;** 4. **O programa deve retornar o valor total após o desconto;** 5. **Deve-se utilizar estruturas if, elif e else (EXIGÊNCIA 1 de 1);** 6. **Colocar um exemplo de SAIDA DE CONSOLE de compra de mais de 10 und. (para mostrar que o desconto foi aplicado)**   **Segue o exemplo de SAIDA DE CONSOLE:**    **OBS: Para os números decimais ficarem com somente duas casas depois da vírgula utilize {:.2f). Exemplo:**  print('O valor sem desconto foi: R$ {:.2f}' .format(sub\_total)) |

| **Apresentação do Código (FORMATO TEXTO)** |
| --- |
| ru = 4467027  print(f'Olá, Bem vindo(a) à calculadora de descontos do Elias Inácio Chavier Neto! ({ru})\n')  while True: *# repete enquanto não receber um valor válido*  try:  valor = float(input('Informe o valor do produto: ')) *# valor do produto*  except:  print('\nInválido. Digite novamente\n')  continue *# reinicia caso valor seja inválido*  break *# finaliza o loop caso o valor seja válido*  while True:  try:  quandade = int(input('Quantidade do produto: ')) *# quantidade do produto*  except:  print('\nInválido. Digite novamente.\n')  continue  break  totalSemDesconto = valor \* quandade *# valor total sem desconto*  if (quandade <= 9): *# validação do desconto baseando na quantidade*  desconto = 0  elif (quandade <= 99):  desconto = 5  elif (quandade <= 999):  desconto = 10  else:  desconto = 15  valorDoDesconto = totalSemDesconto \* desconto/100 *# valor do desconto*  total = totalSemDesconto - valorDoDesconto *# valor final com desconto*  print(f'\nSem desconto o valor a pagar seria R$ {totalSemDesconto:.2f} reais.')  print(f'Com o desconto de {desconto}%, o valor a pagar será de R$ {total:.2f} reais.') |

| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| --- |
|  |

| **QUESTÃO 2 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 04** |
| --- |
| **Enunciado: Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma lanchonete. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.**  **A lanchonete possui seguinte tabela de produtos listados com sua descrição, códigos e valores:**   | Código | Descrição | Valor(R$) | | --- | --- | --- | | 100 | **Cachorro-Quente** | **9,00** | | 101 | **Cachorro-Quente Duplo** | **11,00** | | 102 | **X-Egg** | **12,00** | | 103 | **X-Salada** | **13,00** | | 104 | **X-Bacon** | **14,00** | | 105 | **X-Tudo** | **17,00** | | 200 | **Refrigerante Lata** | **5,00** | | 201 | **Chá Gelado** | **4,00** |   **Elabore um programa em Python que:**   1. **Entre com o código do produto desejado;** 2. **Pergunte se o cliente quer pedir mais alguma coisa (se sim repetir o passo item 2. Caso contrário ir para próximo passo);** 3. **Encerre a conta do cliente com o valor total;** 4. **Deve-se utilizar estruturas if, elif e else (EXIGÊNCIA 1 de 3);** 5. **Se a pessoa digitar um NÚMERO diferente dos da tabela printar na tela: ‘opção inválida’ e voltar para o menu (EXIGÊNCIA 2 de 3);** 6. **Deve-se utilizar while, break, continue (EXIGÊNCIA 3 de 3);**    * **(DICA: utilizar o continue dentro else que verifica a opção inválida)**    * **(DICA: utilizar o break dentro elif que verifica a opção sair)** 7. **Colocar um exemplo de SAIDA DE CONSOLE com dois pedidos** 8. **Colocar um exemplo de SAIDA DE CONSOLE com erro ao digitar no pedido**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 2 itens pedido (Sendo que um deles teve uma tentativa com erro) e no final o valor final é apresentado.** |

| **Apresentação do Código (FORMATO TEXTO)** |
| --- |
| *# variáveis globais*  ru = 4467027  total = 0  pedidoAberto = True  print(f'Olá, Bem vindo(a) à lanchonete do Elias Inácio Chavier Neto! ({ru})')  print('Insira o código do produto para adicioná-lo ao pedido.')  print("""  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  |\_\_\_\_\_\_\_\_\_>>\_TABELA\_DE\_PREÇOS\_<<\_\_\_\_\_\_\_\_\_\_|  |\_\_Cod.\_|\_\_\_\_\_\_Descrição\_\_\_\_\_\_\_\_\_|\_\_Valor\_|  |\_\_100\_\_|\_Cachorro-Quente\_\_\_\_\_\_\_\_|\_\_\_9,00\_|  |\_\_101\_\_|\_Cachorro-Quente\_Duplo\_\_|\_\_11,00\_|  |\_\_102\_\_|\_X-Egg\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_12,00\_|  |\_\_103\_\_|\_X-Salada\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_12,00\_|  |\_\_104\_\_|\_X-Bacon\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_14,00\_|  |\_\_105\_\_|\_X-Tudo\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_17,00\_|  |\_\_200\_\_|\_Refrigerante\_Lata\_\_\_\_\_\_|\_\_\_5,00\_|  |\_\_201\_\_|\_Chá\_Gelado\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_4,00\_|  """)  while pedidoAberto: *# enquanto o pedido estiver em aberto, recebe os itens e adiciona ao pedido*  opt = input('Digite o código do lanche: ') *# código do lanche*  print('')  *# valida a opção recebida e soma o valor do lanche ao total*  if (opt == '100'):  print('1 Cachorro-Quente adicionado ao seu pedido (+ R$ 9,00)')  total += 9  elif (opt == '101'):  print('1 Cachorro-Quente Duplo adicionado ao seu pedido (+ R$ 11,00)')  total += 11  elif (opt == '102'):  print('1 X-Egg adicionado ao seu pedido (+ R$ 12,00)')  total += 12  elif (opt == '103'):  print('1 X-Salada adicionado ao seu pedido (+ R$ 13,00)')  total += 13  elif (opt == '104'):  print('1 X-Bacon adicionado ao seu pedido (+ R$ 14,00)')  total += 14  elif (opt == '105'):  print('1 X-Tudo adicionado ao seu pedido (+ R$ 17,00)')  total += 17  elif (opt == '200'):  print('1 Refrigerante Lata adicionado ao seu pedido (+ R$ 5,00)')  total += 5  elif (opt == '201'):  print('1 Chá Gelado adicionado ao seu pedido (+ R$ 4,00)')  total += 4  else:  *# caso a opção não exista, volta para o inícioe questiona nova,eente o código*  print('Código inválido. Por favor, digite novamente. \n')  continue  print('')  *# pergunta se vai encerrar, se a resposta for válida decide se encerra ou não*  *# se a resposta não for correta repete a pergunta*  while True:  opt2 = input('Deseja mais alguma coisa? [S] Sim / [N] Não ')  if (opt2 == 'S' or opt2 == 's'):  pedidoAberto = False  break  elif (opt2 == 'N' or opt2 == 'n'):  pedidoAberto = True  break  else:  print('\nErro. Tente novamente. \n')  continue    print('')  if pedidoAberto:  break  *# retorna o valor total do pedido ao usuário*  print(f'Total do seu pedido: R$ {total:.2f}'.replace('.', ',')) |

| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| --- |
|  |
| **QUESTÃO 3 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 05** |
| **Enunciado: Imagina-se que você e sua equipe foram contratados por uma empresa de logística que acabou de entrar no ramo. Essa empresa trabalha com encomendas de pequeno e médio porte e opera somente entre 3 cidades.**  **O valor que a empresa cobra por objeto é dado pela seguinte equação:**  **que você e sua equipe**  **Em que cada uma das variáveis que compõe o preço total é quantizada da seguinte maneira:**     | **Quadro 1: Dimensões versus Valor**   | dimensões (cm³) | valor (R$) | | --- | --- | | volume < 1000 | **10** | | 1000 <= volume < 10000 | **20** | | 10000 <= volume < 30000 | **30** | | 30000 <= volume < 100000 | **50** | | volume >= 100000 | **Não é aceito** | | **Quadro 2: Peso versus multiplicador**   | peso(kg) | multiplicador | | --- | --- | | peso <= 0.1 | **1** | | 0.1 <= peso < 1 | **1.5** | | 1 <= peso < 10 | **2** | | 10 <= peso < 30 | **3** | | peso => 30 | **Não é aceito** | | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Quadro 3: Rota versus multiplicador**   | rota | multiplicador | | --- | --- | | RS - De Rio de Janeiro até São Paulo | **1** | | SR - De São Paulo até Rio de Janeiro | **1** | | BS - De Brasília até São Paulo | **1.2** | | SB - De São Paulo até Brasília | **1.2** | | BR - De Brasília até Rio de Janeiro | **1.5** | | RB - Rio de Janeiro até Brasília | **1.5** |   **Obs.: Pode-se mudar o nome das cidades e siglas. Utilizando 3 cidades está ótimo** | |   **Elabore um programa em Python que:**   1. **Pergunte a altura (em cm), comprimento (em cm) e largura (em cm) do objeto. Se digitar um valor não numérico e/ou as dimensões passarem do limite aceito repetir a pergunta;** 2. **Pergunte o peso do objeto (em kg). Se digitar um valor não numérico e/ou o peso passar do limite aceito repetir a pergunta;** 3. **Pergunte a rota do objeto. Se digitar uma opção que não esteja na tabela repetir a pergunta;** 4. **Encerre o total a ser pago com base na equação desse enunciado;** 5. **Deve-se codificar uma função dimensoesObjeto (EXIGÊNCIA 1 de 3);**    * **Dentro da função perguntar altura do objeto (em cm);**    * **Dentro da função perguntar o comprimento do objeto (em cm);**    * **Dentro da função perguntar a largura do objeto (em cm)**    * **Calcular o volume (em cm) da caixa p/a objeto (altura\*largura\*comprimento);**    * **Deve-se ter try/except para o caso do usuário digitar um valor não numérico;**    * **Deve-se retornar o valor em (RS) conforme a Quadro 1** 6. **Deve-se codificar uma função pesoObjeto (EXIGÊNCIA 2 de 3);**    * **Dentro da função perguntar peso do objeto (em kg);**    * **Deve-se ter um try/except para o caso de o usuário digitar um valor não numérico;**    * **Deve-se retornar o multiplicador conforme o Quadro 2** 7. **Deve-se codificar uma função rotaObjeto (EXIGÊNCIA 3 de 3);**    * **Dentro da função perguntar a rota do objeto desejada (Sugestão: utilize as siglas para facilitar os testes);**    * **OBS: PODE MUDAR O NOME DAS CIDADES E SUAS SIGLAS**    * **Deve-se retornar o multiplicador conforme o Quadro 3** 8. **Colocar um exemplo de SAIDA DE CONSOLE uma encomenda com peso, dimensões e rota válidos** 9. **Colocar um exemplo de SAIDA DE CONSOLE com o tratamento de erro quando digitado um valor não numérico é digitado no campo peso ou dimensões)**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com tratamento de valor e erro de digitação** |
| **Apresentação do Código (FORMATO TEXTO)** |
| ru = 4467027  print(f'Olá, Essa é a calculadora de valores do Elias Inácio Chavier Neto! {ru}\n')  rotas = """  Cód Rota  [RS] - De Rio de Janeiro até São Paulo  [SR] - De São Paulo até Rio de Janeiro  [BS] - De Brasília até São Paulo  [SB] - De São Paulo até Brasília  [BR] - De Brasília até Rio de Janeiro  [RB] - Rio de Janeiro até Brasília"""  def dimensoesObjeto():  *# Pergunta ao usuário as dimensões, calcula o volume e retorna a taxa*  while True:  dimensoes = {'altura': 0, 'comprimento': 0, 'largura': 0} *# dimensões*  *# para cada dimensão, mostra o id referente e recebe seu valor.*  *# caso o usuário digite algo incorreto, repete.*  for d in dimensoes:  while True:  try:  dimensoes[d] = float(input(d + ' (cm): '))  break  except:  print('\nValor inválido. Por favor digite novamente.\n')  *# calcula o volume*  volume = dimensoes['altura'] \* dimensoes['comprimento'] \* dimensoes['largura']  print(f'\nVolume do seu objeto: {volume:.1f} cm³')  *# retorna a taxa, caso o volume seja válido*  *# repete as questões caso o volume exceda*  if (volume < 1000):  return 10  elif (volume < 10000):  return 20  elif (volume < 30000):  return 30  elif (volume < 100000):  return 50  else:  print('\nDimensões excedentes. O volume não pode ultrapassar 100000.\n')  continue  def pesoObjeto():  *# Pergunta ao usuário o peso e retorna o multiplicador*  *# caso esteja incorreto ou inválido, repete*  while True:  try:  peso = float(input('peso (kg): '))  except:  print('\nPeso inválido. Por favor digite novamente.\n')  continue  print(f'\nPeso do seu objeto: {peso:.1f} kg')  if (peso < 0.1):  return 1  elif (peso < 1):  return 1.5  elif (peso < 10):  return 2  elif (peso < 30):  return 3  else:  print('\nPeso inválido. Por favor digite novamente.\n')  continue  def rotaObjeto():  *# Pergunta ao usuário a rota e retorna o multiplicador*  *# caso a rota seja inválida, repete*  while True:  print(rotas)  rota = input('\nrota: ')  *# tratativa caso usuário digite correto mas em letra minúscula*  rota = rota.upper()    if (rota == 'RS' or rota == 'SR'):  return 1  elif (rota == 'BS' or rota == 'SB'):  return 1.25  elif (rota == 'BR' or rota == 'RB'):  return 1.5  else:  print('\nRota inválida. Por favor digite novamente. ')  continue  print('Primeiro informe as dimensões do objeto. \n')  valor = dimensoesObjeto()  print('\nAgora informe o peso do objeto. \n')  multiplicador1 = pesoObjeto()  print('\nAgora informe a rota. Exemplo: SR')  multiplicador2 = rotaObjeto()  *# calcula o valor total e mostra*  total = valor \* multiplicador1 \* multiplicador2  print(f'\nTotal a pagar: R$ {total:.2f} (valor: {valor} \* peso: {multiplicador1} \* rota: {multiplicador2})') |

| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| --- |
|  |

| **QUESTÃO 4 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 06** |
| --- |
| **Enunciado: Imagina-se que você está desenvolvendo um software de controle de estoque para uma bicicletaria. Este software deve ter o seguinte menu e opções:**   1. **Cadastrar Peça** 2. **Consultar Peça**    1. **Consultar Todas as Peças**    2. **Consulta Peças por Código**    3. **Consulta Peças por Fabricante**    4. **Retornar** 3. **Remover Peça** 4. **Sair**   **Elabore um programa em Python que:**   1. **Deve-se codificar uma função cadastrarPeca (código) (EXIGÊNCIA 1);**    * **Essa função recebe como parâmetro um código exclusivo para cada peça cadastrado (DICA: utilize um contador como parâmetro)**    * **Dentro da função perguntar o nome da peça;**    * **Dentro da função perguntar o fabricante da peça;**    * **Dentro da função perguntar o valor da peça**    * **Cada peça cadastrada deve ter os seus dados armazenados num DICIONÁRIO (DICA: Conferir material escrito da p. 22 até p24 da AULA 06)** 2. **Deve-se codificar uma função consultarPeca(EXIGÊNCIA 2);**    * **Dentro da função ter um menu com as seguintes opções:**      + **Consultar Todas as Peças**      + **Consultar Peças por Código**      + **Consultar Peças por Fabricante**      + **Retornar** 3. **Deve-se codificar uma função chamada removerPeca (EXIGÊNCIA 3);**    * **Dentro da função perguntar qual o código do produto que se deseja remover do cadastro (da lista de dicionário)** 4. **Colocar um exemplo de SAIDA DO CONSOLE com o cadastro de 3 (ou mais) peças. Sendo que 2 delas do mesmo fabricante – ver figura 1** 5. **Colocar um exemplo de SAIDA DO CONSOLE com a consulta a todas as peças cadastradas – ver figura 2** 6. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por código – ver figura 3** 7. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por fabricante – ver figura 4** 8. **Colocar um exemplo de SAIDA DO CONSOLE ao remover um cadastro e mostrando depois todos os cadastros– ver figura 5**   **Segue os exemplos de SAIDA DE CONSOLE:**    **Figura 1: Exemplo de Cadastrar de três peças. Perceba que 2 delas tem o mesmo fabricante.**    **Figura 2: Exemplo de Consulta Todas as Peças.**    **Figura 3: Exemplo de Consultar por Código.**    **Figura 4: Exemplo de Consultar Peças por Fabricante**    **Figura 5: Exemplo de Remover Peça do cadastro e depois Consultar Todas as Peças. Veja que a peça de código 1 foi removida e não aparece mais no sistema.** |

| **Apresentação do Código (FORMATO TEXTO)** |
| --- |
| ru = 4467027  print('Olá, Esse é o controle de estoque do Elias Inácio Chavier Neto!')  menu1 = '''  ----------------------------------  Insira abaixo a opção desejada:  1 - Cadastrar Peça  2 - Consultar Peça  3 - Remover Peça  4 - Sair  ----------------------------------'''  menu2 = '''  ----------------------------------  Insira abaixo a opção desejada:  1 - Consultar todas as peças  2 - Consultar peças por código  3 - Consultar peças por Fabricante  4 - Retornar  ----------------------------------'''  todasAsPecas = []  numeroDePecasCadastradas = 0 *# usado na geração de código para sempre gerar um diferente*  def cadastrarPeca (): *# cadastra a peça retornada pela função criarPeca*  print('\nCadastro de nova peça:')  peca = criarPeca()  todasAsPecas.append(peca)  print(f"\n{peca['nome']} cadastrada com sucesso. (Código: {peca['codigo']})")  def criarPeca(): *# cria um dicionário padrão para a peça, recebe as informações do usuário e retorna a peça*  *global* numeroDePecasCadastradas  peca = {'nome': '', 'fabricante': '', 'valor': 0}  *for* chave *in* peca: *# para cada key da peca, é associado o valor que o usuário informar*  *while* True:  *try*:  res = input(chave.capitalize() + ': ')  *if* (res == ''): *# trata também como erro uma resposta vazia do usuário*  *raise* Exception("Campo não pode ficar vazio.")  *if* (chave == 'valor'): *# passa para float especificamente no campo 'valor'*  res = float(res)  peca[chave] = res *# associa cada valor a sua chave no dicionário*  *break*  *except*:  print('Valor inválido. Por favor digite novamente.')    numeroDePecasCadastradas += 1 *# geração do código da peça*  codigo = numeroDePecasCadastradas  peca['codigo'] = str(codigo).zfill(3) *# ajuste para todas as peças terem o mesmo padrão '001'*    *return* peca  def consultarPeca(): *# consulta de peças. dispara o menu, recebe as opções, e dispara as funções correspondentes.*  *while* True:  print(menu2)  opt2 = input('> ')  *if* (opt2 == '1'):  mostrarTodasAsPecas()  *elif* (opt2 == '2'):  mostrarPecaPorCodigo()  *elif* (opt2 == '3'):  mostrarPecasPorFabricante()  *elif* (opt2 == '4'):  *break*  *else*:  print('Opção inválida.')  *continue*  def mostrarTodasAsPecas ():  print('\nTodas as peças cadastradas\n')  mostrarPecas(todasAsPecas)  def mostrarPecaPorCodigo ():  codigo = input('\nCódigo: ')  codigo = codigo.zfill(3) *# tratativa para códigos como 1, 01 e 001 serem aceitos*    pecas = []  *for* peca *in* todasAsPecas:  *if* (codigo == peca['codigo']): *# tratativa para códigos como 1, 01 e 001 serem aceitos*  pecas.append(peca)  print(f'\nPeças cadastradas com código {codigo}:\n')  mostrarPecas(pecas)  def mostrarPecasPorFabricante ():  fabricante = input('\nFabricante: ')  fabricante = fabricante.lower()    pecas = []  *for* peca *in* todasAsPecas:  *if* (fabricante == peca['fabricante'].lower()): *# tratativa para códigos como 1, 01 e 001 serem aceitos*  pecas.append(peca)  print(f'\nPeças cadastradas do fabricante {fabricante}:\n')  mostrarPecas(pecas)  def mostrarPecas(pecas): *# função apenas para mostrar as peças passadas como parâmetro. formatando-as e mostrando.*  print('-' \* 78)  print(f" Código | {'Nome':30} | {'Fabricante':20} | {'Valor':5s}")  print('-' \* 78)    *for* peca *in* pecas:  print(f" {peca['codigo']:>6} | {peca['nome']:30} | {peca['fabricante']:20} | R$ {peca['valor']:5.2f}")    *if* (len(pecas) == 0):  print(' Nenhuma peça encontrada.')  print('-' \* 78)  def removerPeca():  codigo = input('\nCódigo da peça: ')  codigo = codigo.zfill(3) *# equaliza códigos como 1, 01 e 001*  *for* peca *in* todasAsPecas: *# procura a peça e remove-a*  *if* (peca['codigo'] == codigo):  x = todasAsPecas.index(peca)  todasAsPecas.pop(x)  print('\nPeça removida.\n')  *# programa principal com o menu principal*  *while* True:  print(menu1)  opt = input('> ')  *if* opt == '1':  cadastrarPeca()  *continue*  *elif* opt == '2':  consultarPeca()  *elif* opt == '3':  removerPeca()  *elif* opt == '4':  print('Encerrando...')  *break*  *else*:  print('\nOpção inválida') |

| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| --- |
| 4. cadastro de 3 peças:    5. consulta de todas as peças:    6. consultas por código:    7. consulta por fabricante:    8. remoção de cadastro: |