

ATIVIDADE PRÁTICA:

Lógica de Programação e Algoritmos

**ANO**

**2023**

Aluno:

WESLEY ALEXANDRE RODRIGUES MARTINS

|  |
| --- |
| **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:**    Colocar o seu nome  **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)** |
| # Identificador Pessoal: print("Seja bem vindo a loja do Wesley Alexandre Rodrigues Martins");  # Recebendo valor unitário do produto: valorProduto = float(input("Digite o valor do produto unitário: "));  # Recebendo a quantidade de produtos: qntdProduto = int(input("Digite a quantidade de produtos que deseja comprar: "));  # Calculando o valor sem desconto: valorSemDesconto = valorProduto \* qntdProduto; # Desconto para compras com o total de até 9 produtos unitários: if qntdProduto <= 9:  valorDesconto = 0; # Desconto para compras com o minímo de 10 e até o máximo de 99 produtos unitários: elif qntdProduto >= 10 and qntdProduto <= 99:  valorDesconto = 0.05; # Desconto para compras com o minímo de 10 e até o máximo de 99 produtos unitários: elif qntdProduto >= 100 and qntdProduto <= 999:  valorDesconto = 0.10; # Desconto para compras com o valor superior à 999 produtos unitários: else:  valorDesconto = 0.15; #Calculando o valor aplicando os descontos: valorComDesconto = valorSemDesconto - (valorSemDesconto \* valorDesconto);  #Retornando os valores sem desconto e com desconto ao usuário: print("O valor total do seu pedido SEM DESCONTO é: R$ {:.2f}" .format(valorSemDesconto)); print("O valor total do seu pedido COM DESCONTO é: R$ {:.2f}" .format(valorComDesconto, valorDesconto \* 100)); |

|  |
| --- |
| **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)** |
| # Identificador Pessoal: print("Seja bem vindo a lanchonete do Wesley Alexandre Rodrigues Martins!");  # Tabela de produtos listados com sua descrição, código e valores: print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cardápio\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); print("| Código | Descrição | Valor (R$)|"); print("| 100 | Cachorro-Quente | R$ 9,00 |"); print("| 101 | Cachorro-Quente Duplo | R$ 11,00 |"); print("| 102 | X-Egg | R$ 12,00 |"); print("| 103 | X-Salada | R$ 13,00 |"); print("| 104 | X-Bacon | R$ 14,00 |"); print("| 105 | X-Tudo | R$ 17,00 |"); print("| 200 | Refrigerante Lata | R$ 5,00 |"); print("| 201 | Chá Gelado | R$ 4,00 |");  # Listagem dos produtos com os códigos na tabela: listaProdutos = {  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: ("Refirgerante Lata", 5.00),  201: ("Chá Gelado", 4.00), }  # Realizando os calculos do pedido totalPedido = 0; while True:  codigoProduto = int(input("Digite o código do produto: "));  if codigoProduto not in listaProdutos:  print("O código digitado é inválido.");  continue;   produto = listaProdutos[codigoProduto];  descricaoProduto = produto[0];  valor = produto[1];  totalPedido += valor;  # Retornando as informações do pedido para o usuário, com a opção de continuar:  print(f"Seu pedido é {descricaoProduto}. O total é {valor:.2f}");  acrescentarPedido = int(input("Deseja pedir algo mais?\n1 - Sim\n0 - Não\n>>"));   if acrescentarPedido == 0:  print("Finalizando o pedido...");  break;  print(f"O valor total do pedido é: R$ {totalPedido:.2f}"); |

|  |
| --- |
| **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:**  Colocar o seu nome    **Figura: Exemplo de programa com tratamento de valor e erro de digitação** |
| **Apresentação do Código (FORMATO TEXTO)** |
| # Identificador Pessoal: print("Bem vindo a Companhia de Logística do Wesley Alexandre Rodrigues Martins S.A!");  # Recebendo as dimensões do objeto: def dimensoesObjeto():  while True:  try:  comprimentoObjeto = float(input("Digite o comprimento do objeto (cm): "));  alturaObjeto = float(input("Digite a altura do objeto (cm): "));  larguraObjeto = float(input("Digite a largura do objeto (cm): "));  # Realizando os calculos com as dimensões informadas:  volumeObjeto = comprimentoObjeto \* alturaObjeto \* larguraObjeto;  print("O volume do objeto é (cm): {}" .format(volumeObjeto));  if volumeObjeto < 1000:  return 10;  elif 1000 <= volumeObjeto < 10000:  return 20;  elif 10000 <= volumeObjeto < 30000:  return 30;  elif 30000 <= volumeObjeto < 100000:  return 50;   # Retornando valores incorretos ou dimensões superiores às listadas:  else:  print("Seu objeto possui dimensões superiores às que trabalhamos. Digite novamente: ");  continue;  except ValueError:  print("Você utilizou um valor não número em algumas das dimensões, favor repetir o processo. Informe as dimensões do produto: "); # Realizando o calculo do peso: def peso():  while True:  try:  pesoObjeto = float(input("Digite o peso do objeto (kg): "));  if pesoObjeto <= 0.1:  return 1;  elif 0.1 < pesoObjeto <= 1:  return 1.5;  elif 1 <= pesoObjeto < 10:  return 2;  elif 10 <= pesoObjeto < 30:  return 3;  # Retornando valores incorretos:  else:  print("Seu objeto possui um peso superior ao que trabalhamos.");  except ValueError:  print("Você digitou um valor não númerico, digite novamente o peso do produto (kg): ");  # Calculando o valor da rota do objeto: def valorRotaObjeto():  while True:  rotaObjeto = input("Selecione a rota:\nRS - Rio de Janeiro até São Paulo\nSR - São Paulo até Rio de Janeiro\nBS - Brasília até São Paulo\nSB - São Paulo até Brasília\nBR - Brasília até Rio de Janeiro\nRB - Rio de Janeiro até Brasília\n>>");  if rotaObjeto == "RS":  return 1;  elif rotaObjeto == "SR":  return 1;  elif rotaObjeto == "BS":  return 1.2;  elif rotaObjeto == "SB":  return 1.2;  elif rotaObjeto == "BR":  return 1.5;  elif rotaObjeto == "RB":  return 1.5;  # Digitando valores incorretos:  else:  print("Você está digitando valores incorretos, favor verificar e digitar novamente a rota do objeto: ");  # Realizando os calculos finais do objeto: dimensoes = dimensoesObjeto() pesoDoObjeto = peso(); rotaObjeto = valorRotaObjeto(); valorFinal = dimensoes \* pesoDoObjeto \* rotaObjeto;  # Exibindo os valores para o usuário: print("O valor total do envio da encomenda, é: R$ {:.2f}\nOs calculos desse valor em detalhes, são: (Dimensões: {:.0f} \* Peso: {:.0f} \* Rota: {:.1f})" .format(valorFinal, dimensoes, pesoDoObjeto, rotaObjeto)); |

|  |
| --- |
| **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)** |
| # Identificador Pessoal: print("Bem vindo ao Controle de Estoque da Bicicletaria do Wesley Alexandre Rodrigues Martins S.A!");  # Cadastrando as peças: estoque = [] def cadastrarPeca(codigo):  print(f"Código: {codigo}")  nome = input("Digite o nome da peça: ")  fabricante = input("Digite o fabricante da peça: ")  valor = float(input("Digite o valor da peça: "))  peca = {  "codigo": codigo,  "nome": nome,  "fabricante": fabricante,  "valor": valor  }  estoque.append(peca)  #Consultando as peças de forma mais detalhada: def consultarPeca():  while True:  print("Escolha a opção desejada: ")  print("1-Consultar Todas as Peças")  print("2-Consultar Peças por Código")  print("3-Consultar Peças por Fabricante")  print("4-Retornar")  escolha = int(input(">>"))  if escolha == 1:  consultarTodasAsPecas()  elif escolha == 2:  consultarCodigo()  elif escolha == 3:  consultarFabricante()  elif escolha == 4:  break  else:  print("Você escolheu uma opção inválida. Digite novamente: ")  # Consultando todas as pecas: def consultarTodasAsPecas():  for peca in estoque:  print("-" \* 30)  print(f"Código: {peca['codigo']}")  print(f"Nome: {peca['nome']}")  print(f"Fabricante: {peca['fabricante']}")  print(f"valor: {peca['valor']}")  print("-" \* 30)  #Consultando as peças por código: def consultarCodigo():  codigo = int(input("Digite o código da peça: "))  for peca in estoque:  if peca['codigo'] == codigo:  print(f"Código: {peca['codigo']}")  print(f"Nome: {peca['nome']}")  print(f"Fabricante: {peca['fabricante']}")  print(f"valor: {peca['valor']}\n")  return  print("A peça informada não foi encontrada.")  #Consultando as peças por fabricante: def consultarFabricante():  fabricante = input("Digite o nome do fabricante: ")  for peca in estoque:  print('-' \* 30)  if peca['fabricante'] == fabricante:  print(f"Código: {peca['codigo']}")  print(f"Nome: {peca['nome']}")  print(f"Fabricante: {peca['fabricante']}")  print(f"valor: {peca['valor']}\n")  print('-' \* 30)  #Removendo as peças: def removerPeca():  codigo = int(input("Digite o código da peça para ser removida: "))  for peca in estoque:  if peca['codigo'] == codigo:  estoque.remove(peca)  print("A Peça foi removida com sucesso.")  return;  print("A Peça não foi encontrada.")  #Usuário escolhendo o que deseja fazer: while True:  print("Escolha a opção desejada:")  print("1-Cadastrar Peças")  print("2-Consultar Peças")  print("3-Remover Peça")  print("4-Sair")  escolha = int(input(">>"))  #Escolha as opções desejadas:  if escolha == 1:  codigo = len(estoque) + 1;  cadastrarPeca(codigo)  elif escolha == 2:  consultarPeca()  elif escolha == 3:  removerPeca()  elif escolha == 4:  break  else:  print("Você digitou uma opção inválida. Digite novamente.") print("Encerrando o programa.") |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| **Figura 1: Cadastrando três peças. Perceba que 2 delas tem o mesmo fabricante:** |
| **Figura 2: Consulta de Todas as Peças:**    **Figura 3: Consultando por Fabricante:**    **Figura 4: Consultando por Código:**    **Figura 5: Removendo uma peça e consultando todas as peças disponíveis em cadastro:** |