# 1) As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores e lhe contrataram para desenvolver o programa que calculará os reajustes. Elaborar um algoritmo  
# que leia o salário de um colaborador e faça o reajuste segundo o seguinte critério, baseando-se  
# no salário atual.  
# Salários Percentual de Aumento  
# até R$ 280 (inclusive) 20%  
# entre R$ 280 e R$ 700 (inclusive) 15%  
# entre R$ 700 e R$ 1500 (inclusive) 10%  
# acima de R$ 1500 5%  
# Após o aumento ser realizado, deve-se imprimir na tela:  
# • o salário antes do reajuste (em R$);  
# • o percentual de aumento aplicado (em %);  
# • o valor do aumento (em R$);  
# • o novo salário, após o aumento (em R$).  
  
salario = float(input(" Salário do colaborador : "))  
print("=-="\*11)  
if(salario <= 280):  
 percentual = 20  
elif(salario <= 700):  
 percentual = 15  
elif(salario <= 1500):  
 percentual = 10  
else:  
 percentual = 5  
print(" Salário original R$ : ", salario)  
print(" Percentual : ", salario)  
  
percentual = percentual / 100.0  
aumento = percentual \* salario  
novo\_salario = salario + aumento  
  
print(" Aumento R$ : ", salario)  
print(" Novo salário R$ : ", novo\_salario)  
print("=-="\*11)

# 2) Elaborar um algoritmo que leia 12 números binários, isto é, somente serão permitidos  
# números “0s” ou “1s” (deve-se fazer a proteção de dados). Calcule e imprima:  
# • A quantidade de números “1s” e números “0s”.Deve-se também comparar os resultados  
# para saber se existem mais “1s” ou “0s” ou se as quantidades são iguais (deve-se imprimir uma mensagem ao usuário);  
# • O percentual de números “0s” e “1s” digitados.  
def pesquisa\_binaria\_recursiva(A, esquerda, direita, item):  
 *"""Implementa pesquisa binária recursivamente."""* # 1. Caso base: o elemento não está presente.  
 if direita < esquerda:  
 return -1  
 meio = (esquerda + direita) // 2  
 # 2. Nosso palpite estava certo: o elemento está no meio do arranjo.  
 if A[meio] == item:  
 return meio  
 # 3. O palpite estava errado: atualizamos os limites e continuamos a busca.  
 elif A[meio] > item:  
 return pesquisa\_binaria\_recursiva(A, esquerda, meio - 1, item)  
 else: # A[meio] < item  
 return pesquisa\_binaria\_recursiva(A, meio + 1, direita, item)  
  
A = [0, 10, 20, 30, 40, 50, 60, 70]  
print("Pesquisa com sucesso:", pesquisa\_binaria\_recursiva(A, 0, len(A) - 1, 20))  
print("Pesquisa com sucesso:", pesquisa\_binaria\_recursiva(A, 0, len(A) - 1, 0))  
print("Pesquisa com sucesso:", pesquisa\_binaria\_recursiva(A, 0, len(A) - 1, 70))  
print("Pesquisa com sucesso:", pesquisa\_binaria\_recursiva(A, 0, len(A) - 1, 100))

# 3) Elaborar um algoritmo que leia uma quantidade indeterminada de números inteiros maiores  
# ou iguais a 1 e menores ou iguais a 99. O algoritmo deverá fazer a proteção de dados no código.  
# Um número igual a zero (0) encerra o programa (flag). Deverá calcular e imprimir:  
# • A média de todos os números digitados;  
# • O percentual de números maiores ou iguais a 1 e menores ou iguais a 10, entre todos os  
# números digitados;  
# • O menor número ímpar e divisível por 5 digitado. Se nenhum número ímpar e divisível por 5  
# for digitado, deve-se encaminhar uma mensagem ao usuário.  
# • O maior número par digitado. Se nenhum número par for digitado, deve-se encaminhar uma  
# mensagem ao usuário.  
  
resp = 'S'  
soma = quant = media = maior = menor = igual = 0  
while resp in 'Ss' :  
 num = int(input("Digite um númro : "))  
 soma += num  
 quant += 1  
 if(quant == 1) :  
 maior = menor = num = igual  
 else:  
 if(num > maior) :  
 maior = num  
 if(num < menor) :  
 menor = num  
 resp = str(input("Quer continuar ? [S/N] : ")).upper().strip()[0]  
media = soma / quant  
print(f"Você digitou {quant} números e a média {media}")  
print(f"O maior valor foi {maior} e o menor foi {menor} e o igual foi {igual}")

# 4) Uma fruteira está vendendo frutas com a seguinte tabela de preços:  
# Até 5 Kg Acima de 5Kg  
# Morango R$ 2.50 por Kg R$ 2.20 por Kg  
# Maçã R$ 1.80 por Kg R$ 1.50 por Kg  
# Se o cliente comprar mais de 8Kg em frutas ou o valor total da compra ultrapassar R$ 25,00,receberá ainda um desconto de 10% sobre este total.Elaborar um algoritmo para ler a  
# quantidade de morangos e maças adquiridas (em Kg), calcule e imprima o valor a ser pago pelo cliente.  
  
def calcular\_precos():  
 # variável  
 count = 0  
 calculo\_produto = 0  
  
 # dados do produto.  
 dados\_produto = [("morango", 2.50, 2.20), ("maçã", 1.80, 1.50)]  
  
 while True:  
  
 # variavel  
 finalizar = False  
  
 # perguntar qual tipo de produto desejado.  
 produto = input("Por favor, informe o produto desejado(Morango ou Maçã):")  
  
 for x in range(2):  
 # verificar se tem o produto desejado  
 if produto.lower() == dados\_produto[x][0]:  
 # armazenar a posição do produto na variável count.  
 count = x  
 # Inserir o valor booleano True para a variável finalizar, para que possa interromper o loop while.  
 finalizar = True  
 # interromper o loop for caso tenha encontrado o determinado produto.  
 break  
  
 else:  
 if x == 1:  
 # Informa que o valor esta invalido.  
 finalizar = False  
 print("Valor inválido.", produto)  
  
 if finalizar:  
 break  
  
 while True:  
  
 try:  
  
 # obter o peso do produto.  
 peso = float(input("Por favor, informe o peso desejado:"))  
  
 # verificar se o peso está acima de zero.  
 if peso > 0:  
 break  
  
 else:  
 continue  
  
 except ValueError:  
 print("valor Invalido do peso.")  
 continue  
  
 # calculos do produto  
 if peso <= 5 and peso > 0:  
 # calculando o valor do produto com o peso.  
 calculo\_produto = dados\_produto[count][1] \* peso  
  
 elif peso > 5:  
 # calculando o valor do produto com o peso.  
 calculo\_produto = dados\_produto[count][2] \* peso  
 # verificando se o peso é maior que 8kg ou o valor passa de R$25,00.  
 if peso > 8 or calculo\_produto > 25:  
 calculo\_produto = (dados\_produto[count][2] \* peso) - ((dados\_produto[count][2] \* peso) \* 10 / 100)  
  
 print("Valor a pagar:R$%.2f" % calculo\_produto)  
  
  
calcular\_precos()

# 6) Elaborar um algoritmo que leia 12 números inteiros maiores ou iguais a 0 (zero) e menores  
# ou iguais a 50. Deve ser feita a proteção de dados no código. O algoritmo deverá calcular e  
# imprimir:  
# • A média de todos os números;  
# • O maior número ímpar digitado;  
# • O menor número par digitado.  
# Se nenhum número par for digitado, deve-se contornar o erro encaminhando uma mensagem  
# ao usuário “Nenhum número par foi digitado.“. Se nenhum número ímpar for digitado, deve-se  
# também encaminhar uma mensagem ao usuário “Nenhum número ímpar foi digitado.  
  
resp = 'S'  
soma = quant = media = maior = menor = igual = resultado = numero = 0  
impar = 1  
while resp in 'Ss' :  
 num = int(input("Digite um número : "))  
 soma += num  
 quant += 1  
 if(quant == 1) :  
 maior = menor = num = igual = impar  
 else:  
 if(num > maior) :  
 maior = num  
 if(num < menor) :  
 menor = num  
 if(resultado == 0):  
 resultado = numero % 1  
 resp = str(input("Quer continuar ? [S/N] : ")).upper().strip()[0]  
media = soma / quant  
print(f"Você digitou {quant} números e a média {media}")  
print(f"O maior valor foi {maior} e o menor foi {menor} e o igual foi {igual}.")  
print(f"O número {impar} é impar.")