

## Exercícios de Computação – Python – Autor: Luccas

1 - Escreva um programa em Python que solicite ao usuário a entrada de dois números inteiros. O programa deve então calcular e exibir:

1. A soma dos dois números.
2. A subtração do primeiro número pelo segundo.
3. O produto dos dois números.
4. O quociente da divisão do primeiro número pelo segundo.

**Observação:** Caso o segundo número seja zero, o programa deve exibir uma mensagem indicando que a divisão não é possível.

Utilize as variáveis num1 e num2 para armazenar os valores digitados pelo usuário. Em seguida, atribua os resultados dos cálculos às variáveis A (soma), sub (subtração), produto (multiplicação) e quoc (divisão). Exiba os resultados com mensagens apropriadas, como mostrado na imagem acima.

Resposta:

```
1 num1 = int(input("Digite o primeiro número: "))
2 num2 = int(input("Digite o segundo número: "))
3
4 soma = num1 + num2
5 sub = num1 - num2
6 produto = num1 * num2
7 quoc = num1/num2
8
9 print("Digite o resultado da soma: ", soma)
10 print("Digite o resultado da subtração:", sub)
11 print("Digite o resultado do produto: ", produto)
12 print("Digite o resultado do quociente:", quoc)
```

Resposta com mensagem:

```
1 num1 = int(input("Digite o primeiro número: "))
2 num2 = int(input("Digite o segundo número: "))
3
4 if num1 == 0 or num2 == 0:
5     print("Erro: O número não pode ser zero, tente novamente com outro número.")
6 else:
7     soma = num1 + num2
8     sub = num1 - num2
9     quoc = num1 * num2
10    div = num1 / num2
11
12    print("O Resultado da Soma é:", soma)
13    print("O Resultado da Subtração é:", sub)
14    print("O Resultado do Quociente é:", quoc)
15    print("O Resultado da Divisão é:", div)
```

2 - Escreva um programa em Python que calcule uma função quadrática  $f(x) = ax^2 + bx + c$ . O programa deve solicitar ao usuário que insira os valores dos coeficientes a, b, c e então calcular o valor de Y Vértice usando a formula de Y vértice =  $-\Delta / 4a$ . O seu programa deve exibir o valor de Y Vértice com quatro casas decimais.

Resposta:

```
1  print("f(x) = a*x*x + b*x + c")
2
3  a = float(input("Digite o valor de a: "))
4  b = float(input("Digite o valor de b: "))
5  c = float(input("Digite o valor de c: "))
6
7  delta = b * b - 4 * a * c
8  yv = -delta / (4 * a)
9
10
11 print("O valor do YV é: %.4f" % yv)
12 # No print acima, foi colocado o termo "%.4f %f"
13 # porque no enunciado se pede o YV com 4 casas decimais.
```

3 - Escreva um programa em Python que receba como entrada a arrecadação de impostos de três níveis de governo: municipal, estadual e federal. O programa deve calcular e exibir o percentual que cada um desses valores representa em relação ao total arrecadado.

Especificações:

1. Solicite ao usuário que insira o valor da arrecadação de impostos municipais, estaduais e federais.
2. Calcule o total arrecadado somando os três valores.
3. Calcule o percentual de cada tipo de imposto em relação ao total.
4. Exiba o percentual de cada arrecadação formatado com duas casas decimais.

**Exemplo de Entrada:**

- Impostos municipais: 2000.00
- Impostos estaduais: 3000.00

- Impostos federais: 5000.00

Resposta:

```
1 m = float(input("Digite a arrecadação de impostos municipais: "))
2 e = float(input("Digite a arrecadação de impostos estaduais: "))
3 f = float(input("Digite a arrecadação de impostos federais: "))
4
5 T = m + e + f
6
7 pm = m/(T*100)
8 pe = e/(T*100)
9 pf = f/(T*100)
10
11 print("O total de impostos é: " , T)
12 print("O total de impostos municipais é: %.4f" %pm)
13 print("O total de impostos estaduais é: %.4f" %pe)
14 print("O total de impostos federais é: %.4f" %pf)
15 #Foi colocado o termo "%.4f %pm/pe/pf" para colocar
16 # as quatro casas decimais depois da virgula.
```

4 - Desenvolva um programa em Python que leia dois números inteiros digitados pelo usuário. O programa deve calcular e exibir o quociente e o resto da divisão do primeiro número pelo segundo. Utilize as variáveis num1 e num2 para armazenar os números digitados pelo usuário. Em seguida, realize a divisão inteira e a operação de módulo para calcular o quociente e o resto, respectivamente.

Resposta:

```
1 num1 = int(input("Digite o primeiro número: "))
2 num2 = int(input("Digite o segundo número: "))
3
4 A = num1//num2
5 B = num1%num2
6
7 print("Quociente:", A) # Resposta da Divisão
8 print("Resto: ", B) |
```

5 - Escreva um programa em Python que solicite ao usuário a entrada de um número inteiro de três dígitos. Em seguida, o programa deve calcular a soma dos algarismos individuais desse número e exibir o resultado.

Por exemplo, se o usuário digitar o número **253**, o programa deve calcular **2 + 5 + 3 = 10** e exibir "Resultado da soma dos algarismos: 10".

O código do programa pode ser estruturado da seguinte forma:

1. Leia o número de três dígitos digitado pelo usuário.
2. Separe cada algarismo do número (centena, dezena e unidade).
3. Calcule a soma dos três algarismos.
4. Exiba o resultado da soma.

**Observação:** Considere que o número de entrada sempre terá três dígitos (entre 100 e 999).

Resposta:

```
1  num = int(input("Digite um número com 3 algarismos: "))
2
3  alg = num//100 # Nesse caso o numero 253 é dividido por 100 e o resultado é 2
4  r = num%100 # o num%100 é responsável por realizar uma divisão com resto
5  # No caso 253/100 é igual a 2 com 53 de resto.
6  alg2 = r//10 # R/10 é igual a 5
7  alg3 = r%10 # R%10 é igual a 3
8  soma = alg + alg2 + alg3
9
10 print("Resultado da soma dos algarismos: ", soma)
```

6 - Uma empresa precisa embalar parafusos em três tipos de caixas: Caixas Grandes com capacidade para 250 parafusos, Caixas Médias com capacidade para 50 parafusos e Caixas Pequenas com capacidade para 10 parafusos. O custo de cada caixa é diferente: uma Caixa Grande custa R\$ 8,50, uma Caixa Média custa R\$ 3,20 e uma Caixa Pequena custa R\$ 1,80.

Dado o número total de parafusos, seu objetivo é determinar a quantidade de cada tipo de caixa necessária para embalar todos os parafusos, o número de parafusos restantes (se houver), e o custo total das caixas utilizadas.

Escreva um programa que receba como entrada o número total de parafusos e exiba:

A quantidade de Caixas Grandes, Caixas Médias e Caixas Pequenas necessárias.

O número de parafusos que restarem sem embalagem.

O Custo Total das caixas utilizadas.

Exemplo de entrada e saída:

Entrada: 1864

Resposta:

```
1  total = int(input("Quantidade total de parafusos:")) #Total = 1864
2
3  cg = total // 250 # Caixa Grande 1864/250 = 7 + resto
4  restocg = total % 250 # Descobrindo o resto = 114
5
6  cm = restocg // 50 # Caixa Média 114/50 = 2 + resto2
7  r2 = restocg % 50 # Descobrindo o resto = 14
8
9  cp= r2 // 10 # Caixa Pequena 14/10 = 1 + resto3
10 resto = r2 % 10 # Descobrindo o resto = 4
11
12 custo = (cg * 8.50) + (cm * 3.20) + (cp * 1.80)
13
14 print("Caixas Grandes:", cg)
15 print("Caixas Médias:", cm)
16 print("Caixas Pequenas:", cp)
17 print("Número de parafusos restantes:", resto)
18 print("Custo Total: R$ %.2f" % custo)
19
20 # A %250 / %50 / %10 é usada para descobrir o resto de uma divisão.
```

7 - Você está estudando o tempo necessário para ler uma enciclopédia que possui um número total de páginas informado pelo usuário. Sabendo que você lê 12 páginas por hora, desenvolva um programa em Python que:

1. Solicite ao usuário o número total de páginas da enciclopédia.
2. Calcule o tempo total necessário para ler todas as páginas em horas, minutos e segundos.
3. Apresente o resultado do tempo de leitura em horas, minutos e segundos formatados.

### Regras:

Considere que 1 hora tem 3600 segundos e 1 minuto tem 60 segundos.

O programa deve dividir o tempo total de leitura em horas inteiras, minutos inteiros e os segundos restantes.

### Entrada:

Número de páginas da enciclopédia: 500

### Saída:

Tempo horas: 41

Tempo minutos: 40

Tempo segundos: 0

Resposta:

```
1  tpag = int(input("total de paginas: "))
2  totals = (tpag*3600)/12
3  qh = totals//3600
4  r = totals%3600
5  qm=r//60
6  qs=r%60
7
8  print(qh)
9  print(qm)
10 print(qs)
```

8 - Escreva um código em Python que receba a média final de um aluno e exiba as mensagens "Parabéns!!!" e "Te espero no Cálculo II" caso a média seja maior ou igual a 5. Caso contrário, nenhuma mensagem deve ser exibida.

Resposta:

```
1  MF = float(input("Digite uma nota:"))
2  if (MF>= 5): # Se a nota for maior ou igual a 5 = Aprovado
3      print("Parabéns!!!")
4      print("Te espero no cálculo II")
```

9 - Escreva um programa em Python para calcular o imposto de renda devido por uma pessoa física, com base no salário mensal informado. Para isso, utilize a tabela abaixo como referência:

salário(R\$)	aliquota(%)	dedução
Até 1499,15	Isento ou 0	
De 1.499,16 até 2.246,75	7,5	112,43
De 2.246,76 até 2.995,70	15,0	280,94
De 2.995,71 até 3.743,19	22,5	505,62
Acima de 3.743,19	27,5	692,78

Imposto = (salário\*alíquota/100)-dedução

Resposta:

```

1  salario = float(input("Entre com o salário: R$"))
2  if (salario < 1499.15):
3      Imposto = 0
4
5  if (salario>=1499.16)and(salario <=2246.75):
6      Imposto = (salario*7.5/100)-112.43
7
8  if (salario>=2246.76)and(salario <=2995.70):
9      Imposto = (salario*15.0/100)-280.94
10
11 if (salario>=2995.71)and(salario <=3743.19):
12     Imposto = (salario*22.5/100)-505.62
13
14 if (salario>3743.19):
15     Imposto = (salario*27.5/100)-692.78
16
17 print("Imposto: R$ %.2f" % Imposto)
18 # %.2f no print significa duas casas depois da virgula.

```

10 - Uma locadora de veículos oferece descontos e cobra multas dependendo da quantidade de dias que um cliente demora para devolver o carro alugado. O programa deve calcular o valor a ser pago pelo cliente, com base nas seguintes regras:

- Se o carro for devolvido em até 8 dias, o cliente terá um desconto de 2% sobre o valor do aluguel.
- Se o carro for devolvido entre 9 e 15 dias, não há desconto nem acréscimo no valor do aluguel.
- Se o carro for devolvido após 15 dias, será cobrada uma multa de 20% sobre o valor do aluguel, acrescida de um adicional de R\$ 0,0086 por dia excedente após o 15º dia.

## Requisitos:

O programa deve solicitar dois valores ao usuário:

- O valor do aluguel diário (em reais).
- A quantidade total de dias que o cliente manteve o carro alugado.

Resposta:

```
1  aluguel = float(input("Entre aluguel: "))
2  dia = int(input("Entre dia: "))
3
4  if (dia < 9):
5      valor = aluguel * 0.98
6  elif (dia >= 9) and (dia <= 15): # elif significa uma multipla escolha de if
7      valor = aluguel
8  else:
9      valor = aluguel * 1.2 + aluguel * (dia - 15) * 0.0086
10
11  print("Valor a ser pago: ", valor)
```

11 - Você trabalha em uma fábrica que monta placas de circuito eletrônico utilizando três tipos diferentes de peças: A, B e C. Cada placa é composta por:

- 2 unidades do tipo A,
- 3 unidades do tipo B,
- 7 unidades do tipo C.

O objetivo é determinar a quantidade máxima de placas que podem ser produzidas com os estoques disponíveis de cada tipo de peça.

Resposta:

```
1  Qa = int(input("Qtde de peças A: "))
2  Qb = int(input("Qtde de peças B: "))
3  Qc = int(input("Qtde de peças C: "))
4
5  La = Qa // 2
6  Lb = Qb // 3
7  Lc = Qc // 7
8
9  menor = La
10 if (Lb < menor):
11     menor = Lb
12 if (Lc < menor):
13     menor = Lc
14
15  print("A máxima qtde de placas é", menor)
```

12 - Em um supermercado, o setor de artigos para limpeza deseja oferecer durante uma promoção, um kit de produtos com o preço diferenciado. O kit deve ter quatro unidades de detergente, três de sabão em pó e cinco unidades de esponja. Conhecendo-se as



quantidades de unidades disponíveis de cada um desses itens, como determinar a quantidade de kits que poderão ser colocados à venda?

Considere que todas as quantidades serão expressas por valores inteiros.

**Exemplo:**

Quantidade de detergente: 1555

Quantidade de sabão em pó: 1910

Quantidade de esponja: 4810

Quantidade de kit: 388

Resposta:

```
1 dtg = int(input("Quantidade de detergente: "))
2 sab = int(input("Quantidade de sabão em pó: "))
3 esp = int(input("Quantidade de esponja: "))
4
5 Qd = dtg // 4
6 Qsab = sab // 3
7 Qesp = esp // 5
8
9 menor = Qd # Igualou menor = Quantidade de Detergente
10
11 if Qsab < menor: # Quant de Sabão for menor que Quant Detergente
12     menor = Qsab
13
14 if Qesp < menor: # Quant de Esponja for menor que Quant Detergente
15     menor = Qesp
16
17 print("Quantidade de kit:", menor)
```

13 - Uma determinada marca de televisores apresenta duas linhas de TVs (série 100 e a série 200). A distância ideal entre o sofá e o televisor depende do tamanho da tela e também da própria série, conforme as tabelas abaixo:

Série 100	
Distância (metros)	Tamanho da Tela (Polegadas)
Até 1,4	32
De 1,5 a 2,6	37
Acima de 2,6	42

Série 200	
Distância (metros)	Tamanho da Tela (Polegadas)
Até 2,8	42
De 2,9 a 3,6	50
Acima de 3,6	61

Conhecendo-se a distância e a série pretendida por certo cliente, obtenha o tamanho ideal de sua televisão.

Digite a distância (m): 3.1

Digite a série: 200

Tamanho da televisão (pol): 50

Resposta:

```
1 D = float(input("Digite a distância (m): "))
2 S = float(input("Digite a série: "))
3
4 if(S == 100):
5     if D <= 1.4:
6         print("Tamanho da televisão (pol): 32")
7     if D > 1.5 and D <= 2.6:
8         print("Tamanho da televisão (pol): 37")
9     if D >= 2.6:
10        print("Tamanho da televisão (pol): 42")
11
12 if(S == 200):
13     if D <= 2.8:
14         print("Tamanho da televisão (pol): 42")
15     if D > 2.9 and D <= 3.6:
16         print("Tamanho da televisão (pol): 50")
17     if D > 3.6:
18         print("Tamanho da televisão (pol): 61")
```

14 - "Faça um programa que leia um valor N inteiro e positivo, calcule e mostre o valor de E, conforme a fórmula a seguir:

$$E = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

Resposta:

```
1 N = int(input("Digite o valor de N: "))
2
3 cont = 1
4 soma = 1
5
6 while cont <= N:
7     soma = soma + (1 / cont)
8     cont = cont + 1
9
10 print("O valor de E = %.4f" % soma)
```

15 - Carlos e João decidiram investir parte de seus salários em aplicações financeiras com rendimentos diferentes. Carlos aplica todo o seu salário em uma caderneta de poupança que rende 2% ao mês, enquanto João, que possui um

salário equivalente à metade do de Carlos, investe todo o seu dinheiro em um fundo de renda fixa com rendimento de 25% ao mês.

Escreva um programa em Python que receba o salário inicial de Carlos como entrada e calcule o número de meses necessários para que o montante acumulado por João ultrapasse ou iguale o montante acumulado por Carlos. O programa deve exibir o número de meses necessários para que isso aconteça. Considere que os rendimentos sejam compostos mensalmente (ou seja, os juros são aplicados sobre o saldo atualizado a cada mês).

Resposta:

```
1  c = float(input('Digite o salário de Carlos: R$ '))
2
3
4  j = c / 2 # Salário de João é a metade do salário de Carlos
5
6  # Rendimento mensal de Carlos e João
7  rendimento_carlos = 0.02 # 2% ao mês
8  rendimento_joao = 0.25 # 25% ao mês
9
10 # Inicializa os meses
11 meses = 0
12
13 # Loop até o valor de João ultrapassar o de Carlos
14 while j <= c:
15     c += c * rendimento_carlos # Atualiza o valor de Carlos
16     j += j * rendimento_joao # Atualiza o valor de João
17     meses += 1
18
19 # Exibe o número de meses
20 print("Tempo necessário: meses", meses)
```

16 - Elabore um programa em Python que permita ao usuário inserir uma nota. A nota deve estar no intervalo de 0 a 10. Caso o usuário insira um valor fora desse intervalo, o programa deverá exibir uma mensagem informando que a nota está fora da faixa permitida e solicitará que o usuário insira a nota novamente. O programa deve repetir esse processo até que o usuário forneça uma nota válida.

Resposta:

```
1  nota = float(input("Digite sua nota: "))
2  while (nota < 0) or (nota > 10):
3      print("Nota fora da faixa: ")
4      nota = float(input("Digite novamente sua nota: "))
```

17 - Foram anotadas as medidas dos diâmetros de várias peças produzidas em um torno. Como obter o valor médio dessas medidas? Não é conhecida previamente a quantidade de peças observadas.

Suponha que após a digitação da medida da última peça observada será digitado o valor zero (o zero indica fim da entrada de dados).

### Exemplo:

#### Entrada:

Digite o valor do diâmetro: 33

Digite o valor do diâmetro: 34

Digite o valor do diâmetro: 35.5

Digite o valor do diâmetro: 34.5

Digite o valor do diâmetro: 0

#### Saída:

Média dos diâmetros = 34.25

### Exemplo:

```
1  k = 0 # Contador de peças. Ele armazena a quantidade de diâmetros informados.
2  soma = 0 # Variável que acumula o valor total da soma dos diâmetros digitados.
3  while True: # Cria um laço infinito para repetir a entrada de valores
4      # até que o usuário digite 0, que encerre o Loop.
5      d = float(input("Digite o valor do diâmetro: "))
6      if d == 0: # Indica quando d = 0, interrompe o código
7          break
8      soma = soma + d # Soma cumulativa de todos os diâmetros digitados até o momento.
9      k = k + 1 # Incrementa o contador de peças para cada valor digitado, exceto o 0.
10
11  media = soma / k
12  print("Média dos diâmetros = %.2f" % media)
```

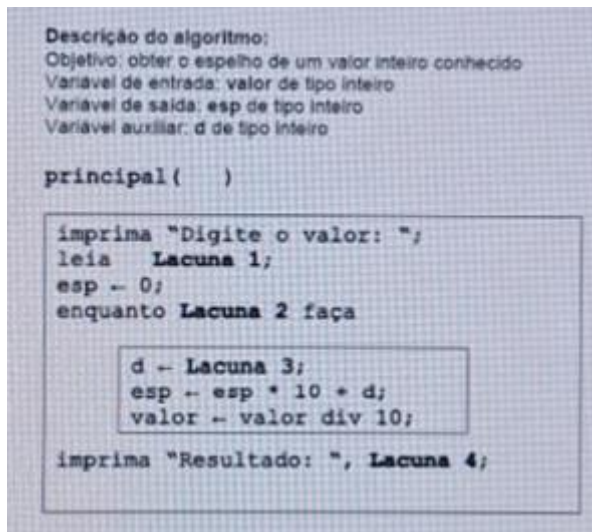
18 - Obter o espelho de um valor inteiro positivo conhecido.

Observação: entenda-se espelho como o valor obtido pela leitura invertida do valor.

Exemplos:

- O espelho de 3629 é 9263.
- O espelho de 301 é 103.
- O espelho de 2000 é 2.
- O espelho de 5 é 5.

O algoritmo incompleto a seguir deve representar um método de resolução do problema.



Resposta:

```
1  valor = int(input("Digite o valor: ")) # Entrada do número inteiro
2  esp = 0                                # Variável que armazenará o espelho do número
3
4  while valor != 0:                       # Enquanto o número ainda tiver dígitos:
5      d = valor % 10                     # Obtém o último dígito do número (resto da divisão por 10)
6      esp = esp * 10 + d                 # Adiciona o dígito ao espelho, "deslocando" os anteriores
7      valor = valor // 10                # Remove o último dígito do número (divisão inteira por 10)
8
9  print("Resultado:", esp)               # Exibe o valor invertido (espelho)
```

19 - "Faça um programa que leia um valor n inteiro e positivo, calcule e mostre o valor de S, conforme a fórmula a seguir:

$$S = 1 + \frac{2}{3} + \frac{3}{5} + \frac{4}{7} + \dots + \frac{n}{2n-1}$$

Resposta:

```

1  # Solicita ao usuário o número de parcelas (n), que deve ser um número inteiro e positivo
2  n = int(input("Digite o número de parcelas:"))
3
4  # Inicializa a variável soma com 0 para acumular o valor da soma das parcelas
5  soma = 0
6
7  # Itera de 1 até n (inclusive), calculando cada parcela da fórmula
8  for i in range(1, n+1):
9      # Adiciona a parcela correspondente (i / (2 * i - 1)) à variável soma
10     soma = soma + i / (2 * i - 1)
11
12 # Exibe o resultado da soma, formatando o número com 5 casas decimais
13 print("A soma das %d primeiras parcelas é %.5f" % (n, soma))

```

20 - Deve-se anotar as medidas (em milímetros) dos diâmetros de 10 peças produzidas em um torno. (Como obter o valor médio dessas medidas que estão entre 50mm e 100mm (inclusive 50mm e 100mm)?)

**Obs.:** Utilize duas casas decimais depois da vírgula.

**Tela:** Apresente na tela com os seguintes valores de entrada:

50, 62.8, 25, 32.8, 39.99, 51.1, 65.84, 79.54, 99, 49.99

**Saída:** A média é 68.05

Resposta:

```

1  # Inicializa a variável 'soma' com 0 para armazenar a soma dos diâmetros válidos.
2  soma = 0
3
4  # Inicializa a variável 'cont' com 0 para contar o número de diâmetros válidos.
5  cont = 0
6
7  # Inicia um loop que irá iterar 10 vezes (de 1 a 10).
8  for a in range(1, 11):
9      # Solicita ao usuário que insira uma medida do diâmetro e converte o valor para ponto flutuante.
10     diam = float(input("Entre medida do diâmetro: "))
11
12     # Verifica se o diâmetro inserido é maior ou igual a 50.
13     if diam >= 50:
14         # Se for válido (>= 50), adiciona o valor à soma.
15         soma = soma + diam
16         # Incrementa o contador de diâmetros válidos.
17         cont = cont + 1
18     else:
19         # Caso contrário, ignora o diâmetro (define diam como 0, embora esta linha não seja usada posteriormente).
20         diam = 0
21
22 # Calcula a média dividindo a soma total dos diâmetros válidos pelo número de valores válidos.
23 med = soma / cont
24
25 # Exibe a média com duas casas decimais.
26 print("A média é %.2f" % med)

```

21 - Faça uma função que receba três notas de um aluno como parâmetros e uma letra. Se a letra for "A", a função deverá calcular a média aritmética das notas do aluno; se for "P", deverá calcular a média ponderada com pesos 5, 3 e 2. A média calculada deve ser devolvida à função principal para, então, ser mostrada.

Exemplo:

Digite a primeira nota: 4.5

Digite a segunda nota: 7.3

Digite a terceira nota: 5.7

Digite A ou P para a letra: A

Média = 5.83

Resposta:

```
1  nota1 = float(input("Digite a primeira nota: "))
2  nota2 = float(input("Digite a segunda nota: "))
3  nota3 = float(input("Digite a terceira nota: "))
4  letra = input("Digite A ou P para a letra: ")
5  # Função para calcular a média aritmética
6  def meda(a, b, c):
7      # Calcula a média aritmética somando as notas e dividindo por 3
8      m = (a + b + c) / 3
9      return m # Retorna o resultado da média aritmética
10
11 # Função para calcular a média ponderada
12 def medp(a, b, c):
13     # Calcula a média ponderada aplicando os pesos 5, 3 e 2 às notas
14     p = (a * 5 + b * 3 + c * 2) / 10
15     return p # Retorna o resultado da média ponderada
16
17 # Verifica se o usuário escolheu calcular a média aritmética
18 if letra == 'A':
19     # Chama a função 'meda' com as notas fornecidas e atribui o resultado a M
20     M = meda(nota1, nota2, nota3)
21     # Exibe o resultado formatado com duas casas decimais
22     print("Média = %.2f" % M)
23
24 # Verifica se o usuário escolheu calcular a média ponderada
25 elif letra == 'P':
26     # Chama a função 'medp' com as notas fornecidas e atribui o resultado a M
27     M = medp(nota1, nota2, nota3)
28     # Exibe o resultado formatado com duas casas decimais
29     print("Média = %.2f" % M)
```

22 - O número 3025 tem a seguinte característica:

$$30 + 25 = 55 \text{ e } 55^2 = 3025.$$

Ou, colocado de forma genérica, um número XYZW pode ter a seguinte característica:

$$(XY) + (ZW) = RS \text{ e } (RS)^2 = XYZW.$$

Como verificar se um número inteiro positivo de quatro algarismos apresenta esta característica.

Se sim, classificar o número como TIPO I.

Se  $(XY) + (ZW) = RS$  e  $(RS)^2 > XYZW$  então classificar o número como TIPO II.

Se  $(XY) + (ZW) = RS$  e  $(RS)^2 < XYZW$  então classificar o número como TIPO III.

Fazer uma verificação na entrada de dados para assegurar que o número tem quatro algarismos e apresentar na saída uma das mensagens:

“TIPO I” ou “TIPO II” ou “TIPO III”.

Construir e utilizar uma função que tenha como parâmetro de entrada um número inteiro positivo de quatro algarismos e como retorno o quadrado da soma descrita acima, ou seja: se a entrada for 3025, devolve para o programa principal o valor 3025, ou se a entrada for 1521, devolve para o programa principal o valor 1296.

O código abaixo apresenta a descrição da função calc, mas está incompleto.

Resposta:

```
1  def calc(num):
2      # Verifica se o número possui exatamente 4 algarismos
3      if num > 1000 or num < 9999: # Correção: Deve ser `1000 <= num <= 9999`
4          a = num // 100 # Divide o número para obter os dois primeiros dígitos (XY)
5          b = num % 100 # Obtém os dois últimos dígitos (ZW)
6          c = a + b      # Soma XY + ZW
7          d = c * c      # Calcula o quadrado da soma (c^2)
8
9      # Verifica o tipo do número conforme as condições dadas
10     if d == num:      # Se o quadrado for igual ao número, é Tipo I
11         return "Tipo I"
12     if d > num:       # Se o quadrado for maior que o número, é Tipo II
13         return "Tipo II"
14     if d < num:       # Se o quadrado for menor que o número, é Tipo III
15         return "Tipo III"
16
17     # Solicita ao usuário que insira um número de 4 algarismos
18     n = int(input("Entre número de 4 algarismos: "))
19
20     # Garante que o número digitado esteja dentro do intervalo permitido (4 algarismos)
21     while n < 1000 or n > 9999:
22         n = int(input("Entre número de 4 algarismos: "))
23
24     # Calcula e imprime o tipo do número usando a função calc
25     print(calc(n))
```



23 - Faça um programa que leia dois vetores de 5 números e armazene num vetor de mesmo tamanho a soma de cada elemento dos dois vetores. Exibir os três vetores.

Resposta:

```
1 # Inicializa três listas vazias: uma para o primeiro vetor, outra para o segundo e uma para armazenar as somas.
2 num1 = [] # Lista para armazenar os números do vetor 1
3 num2 = [] # Lista para armazenar os números do vetor 2
4 soma = [] # Lista para armazenar as somas dos elementos dos dois vetores
5
6 # Solicita ao usuário os números do primeiro vetor.
7 print("Digite 5 números inteiros do vetor 1:")
8 for i in range(5): # Laço que se repete 5 vezes para receber 5 números
9     x = int(input()) # Lê um número inteiro do usuário
10    num1.append(x) # Adiciona o número lido à lista 'num1'
11
12 # Solicita ao usuário os números do segundo vetor.
13 print("Digite 5 números inteiros do vetor 2:")
14 for i in range(5): # Laço que se repete 5 vezes para receber 5 números
15     y = int(input()) # Lê um número inteiro do usuário
16     num2.append(y) # Adiciona o número lido à lista 'num2'
17
18 # Calcula a soma dos elementos correspondentes dos dois vetores.
19 for i in range(5): # Laço que se repete 5 vezes para percorrer os elementos de ambas as listas
20     soma.append(num1[i] + num2[i]) # Soma os elementos de 'num1' e 'num2' no mesmo índice e adiciona à lista 'soma'
21
22 # Exibe os três vetores.
23 print(num1) # Imprime o vetor 1
24 print(num2) # Imprime o vetor 2
25 print(soma) # Imprime o vetor com as somas
```

24 - "As temperaturas em n pontos de uma cidade foram armazenadas em uma lista. Faça um programa que leia as temperaturas em n pontos de uma cidade e que imprima a média das temperaturas e uma lista de pontos da cidade com temperaturas superiores à temperatura média da cidade, em ordem crescente."

Digite o número de pontos: 8

Digite a temperatura 1: -10

Digite a temperatura 2: -8

Digite a temperatura 3: 0

Digite a temperatura 4: 1

Digite a temperatura 5: 2

Digite a temperatura 6: 5

Digite a temperatura 7: -2

Digite a temperatura 8: -4

Temperatura média: -2.00

[3, 4, 5, 6]

Resposta:

```
1  # Inicialização de listas e variáveis
2  temp = [] # Lista para armazenar as temperaturas de cada ponto
3  soma = 0 # Variável para armazenar a soma das temperaturas
4  cidade = [] # Lista para armazenar os índices dos pontos com temperaturas acima da média
5  medial = [] # Lista para armazenar as temperaturas acima da média
6
7  # Solicita ao usuário o número de pontos a serem considerados
8  n = int(input("Digite o número de pontos: "))
9
10 # Loop para coletar as temperaturas dos pontos
11 for i in range(0, n):
12     t = float(input("Digite a temperatura %d: " % (i + 1))) # Lê a temperatura como número decimal
13     temp.append(t) # Adiciona a temperatura à lista 'temp'
14     soma = soma + temp[i] # Soma a temperatura ao total acumulado
15
16 # Calcula a média das temperaturas
17 media = soma / n
18 # Exibe a temperatura média com duas casas decimais
19 print("Temperatura média: %.2f" % media)
20
21 # Loop para identificar pontos com temperaturas acima da média
22 for i in range(0, n):
23     if temp[i] > media: # Verifica se a temperatura do ponto atual é maior que a média
24         medial.append(temp[i]) # Adiciona a temperatura à lista 'medial'
25         medial.sort() # Organiza as temperaturas acima da média em ordem crescente
26         cidade.append(i + 1) # Adiciona o índice do ponto à lista 'cidade' (considerando 1 como o primeiro ponto)
27         cidade.sort() # Organiza os índices dos pontos em ordem crescente
28
29 # Exibe os índices dos pontos com temperaturas acima da média
30 print(cidade)
```

25 - Faça um programa que peça as três notas (entre 0 e 10) de 4 alunos, calcule e armazene numa lista a média aritmética de cada aluno. Imprima a média aritmética de cada aluno e o número de alunos com média maior ou igual a 7.0

Digite 3 notas:

Notas do aluno 1:

1

2

3

Notas do aluno 2:

5

5

6

Notas do aluno 3:

2

2

7

Notas do aluno 4:

10

7

4

[3.0, 5.0, 5.0, 7.0]

Qtde aprovados: 1

Resposta:

```
1 medias = [] # Inicializa uma lista vazia para armazenar as médias dos alunos
2 aprov = 0 # Inicializa o contador de aprovados (alunos com média >= 7.0) como 0
3
4 print("Digite 3 notas: ") # Exibe uma mensagem para orientar o usuário a inserir as notas
5
6 # Laço de repetição para iterar sobre os 4 alunos (range(4) gera valores de 0 a 3)
7 for i in range(4):
8     # Exibe qual aluno está sendo processado (i+1 ajusta o índice para começar do 1)
9     print("Notas do aluno %d: " % (i+1))
10
11     # Solicita as três notas do aluno e converte as entradas de texto para números decimais
12     nota1 = float(input()) # Primeira nota
13     nota2 = float(input()) # Segunda nota
14     nota3 = float(input()) # Terceira nota
15
16     # Calcula a média aritmética das três notas
17     media = (nota1 + nota2 + nota3) / 3
18
19     # Adiciona a média calculada à lista de médias
20     medias.append(media)
21
22 # Imprime a lista de médias de todos os alunos
23 print(medias)
24
25 # Outro laço para verificar quantos alunos têm média maior ou igual a 7.0
26 for i in range(0, 4):
27     # Verifica se a média do aluno atual é maior ou igual a 7.0
28     if (medias[i] >= 7):
29         # Incrementa o contador de aprovados
30         aprov = aprov + 1
31
32 # Imprime a quantidade total de alunos aprovados
33 print("Qtde aprovados: ", aprov)
```

## 26 - PROBLEMA DA ÁREA DO POLÍGONO

Implemente o algoritmo que calcula a área de um polígono simples dado pelo número de vértices e suas coordenadas.

### Exemplo de entrada e saída:

#### PROBLEMA DA ÁREA DO POLÍGONO

Entre com a quantidade de vértices do polígono: 5

Entre coordenada x: 0

Entre coordenada y: 0

Entre coordenada x: 0

Entre coordenada y: 1

Entre coordenada x: 1

Entre coordenada y: 2

Entre coordenada x: 2

Entre coordenada y: 2

Entre coordenada x: 2

Entre coordenada y: 0

Área do polígono = 3.00

Resposta:

```
1  # Inicialização das listas para armazenar as coordenadas dos vértices
2  x = [] # Lista para armazenar as coordenadas x dos vértices
3  y = [] # Lista para armazenar as coordenadas y dos vértices
4  soma = 0 # Variável para acumular a soma parcial usada no cálculo da área
5
6  print("PROBLEMA DA ÁREA DO POLÍGONO") # Impressão inicial do título do problema
7
8  # Solicita ao usuário o número de vértices do polígono
9  qv = int(input("Entre com a quantidade de vértices do polígono: "))
10
11 # Laço para coletar as coordenadas dos vértices
12 for i in range(0, qv, 1): # Para cada vértice do polígono
13     # Solicita e armazena as coordenadas x e y
14     x.append(int(input("Entre coordenada x: "))) # Adiciona o valor de x à lista x
15     y.append(int(input("Entre coordenada y: "))) # Adiciona o valor de y à lista y
16
17 # Laço para calcular a soma parcial dos produtos cruzados dos vértices consecutivos
18 for i in range(0, qv-1, 1): # Itera sobre os vértices, excluindo o último
19     # Soma parcial baseada na fórmula do determinante (produto cruzado)
20     soma = soma + (x[i] + x[i+1]) * (y[i] - y[i+1])
21
22 # Adiciona a última parte da fórmula, que conecta o último vértice ao primeiro
23 soma = soma + (x[qv-1] + x[0]) * (y[qv-1] - y[0])
24 area = soma / 2 # Divide a soma total por 2 para calcular a área do polígono
25
26 print("Área do polígono = %.2f" % area) # Exibe a área do polígono com 2 casas decimais
```

27 - A quantidade de vértices é armazenada na variável qv, as coordenadas x são guardadas na lista x com qv componentes e as coordenadas y são guardadas na lista y de qv componentes.

Complete as lacunas de 1 a 4.

**Entrada:** x e y de tipo int; qv tipo int

**Saída:** área tipo int

```
1 principal ( )
2 # Área Polígono
3 x = []
4 y = []
5 soma = LACUNA 1
6
7 print ("PROBLEMA DA ÁREA DO POLÍGONO")
8 qv = int (input ("Entre com a quantidade de vértices do polígono: "))
9
10 for i in range (LACUNA 2):
11     x.append (int (input("Entre coordenada x: ")))
12     y.append (int (input("Entre coordenada y: ")))
13
14 for i in range (0, qv - 1):
15     soma = soma + (LACUNA 3)
16
17 soma = soma + (x[qv-1] * y[0]) + (y[qv-1] * x[0])
18 area = LACUNA 4
19
20 print ("Área do polígono = %.2f" % area)
```

Resposta:

Observe o algoritmo e preencha as LACUNAS:

- LACUNA 1: 0
- LACUNA 2: 0, qv - 1
- LACUNA 3:  $(x[i] + x[i+1]) * (y[i] - y[i+1])$
- LACUNA 4: soma / 2

28 – Simule a execução do algoritmo abaixo, supondo que as entradas sejam:  
a = 10, b = 9, c = 6 e d = 5.

algoritmoQ1( )

```
entrar a; entrar b; entrar c; entrar d;  
e←0; f←0; g←0;  
se (a>b) ou (a>c) então g←a;  
    senão se (b>d) ou (b<c) então g←b;  
        senão g←c;  
se (a>=b) então f←b; b←a; a←f;  
se (a>b) e (c<d) então e←e+1;  
    senão e←e+2;  
se (b=c) então f←f+2;  
g←g+2;  
se c≠a então g←g+3; e←e+3;  
f←f+3;  
exibir e; exibir f; exibir g;
```

Resposta:

```
1  a = 10  
2  b = 9  
3  c = 6  
4  d = 5  
5  e = 0  
6  f = 0  
7  g = 0  
8  
9  if a > b or a > c:  
10     g = a  
11  
12  elif b>d or b<c:  
13     g = d  
14  else:  
15     g = c  
16  
17  if a>=b :  
18     f = b  
19     b = a  
20     a = f  
21  
22  if a>b and c<d:  
23     e = e + 1  
24  else:  
25     e = e + 2  
26  
27  if b == c:  
28     f = f + 2  
29     g = g + 2  
30  
31  if c!=a:  
32     g = g + 3  
33     e = e + 3  
34     f = f + 3  
35  
36  print(e)  
37  print (e+f)
```

29-Conhecendo-se o valor total do pedido e o número de produtos adquiridos, calcular e exibir o valor a ser pago considerando as seguintes condições:

- Pedidos acima de R\$ 200,00 com um único produto recebem desconto de 8% e frete grátis;
- Pedidos acima de R\$ 200,00 com mais de um produto, recebem desconto de 8% e pagam frete de R\$ 7,00;
- Pedidos que custam entre R\$ 150,00 e R\$ 200,00 (inclusive) recebem desconto de 7% e pagam frete de R\$ 15,00;
- Demais situações não recebem desconto e pagam frete de R\$ 15,00.

Copie e cole o código em Python e a Tela de Execução no quadro de resposta.

Resposta:

```
1  x = (int(input("Digite o valor total de pedidos:" )))
2  y = (int(input("Digite o número de produtos adquiridos: ")))
3
4  if x > 200 and y == 1:
5      frete = 0
6      desconto = 0.08
7
8  if x > 200 and y > 1:
9      frete = 7
10     desconto = 0.08
11
12     if x > 150 and x <= 200:
13         frete = 15
14         desconto = 0.07
15
16     else:
17         frete = 15
18         desconto = 0
19
20     d = (x*desconto)
21     f = frete
22     vf = (x-d+f)
23
24     print("O valor do desconto:", desconto)
25     print("O valor do frete é:", frete)
26     print("O valor final é:", vf)
```

30 - Uma parede retangular precisa receber um revestimento de azulejos.

- Conhecendo-se as dimensões da parede (altura e comprimento) e a dimensão de cada azulejo quadrado (lado).
- Calcular e exibir quantos azulejos serão necessários para recobrir essa parede.
- Os azulejos não serão recortados, dessa forma, uma parcela da parede poderá não ser revestida.
- Determinar também a área da parede que ficará sem revestimento.
- Todas as medidas (altura/comprimento/lado) são dadas por valores inteiros em centímetros.

Altura da parede = 200 cm / Comprimento da parede = 250 cm / Lado do azulejo = 45.

*Total de azulejos = 20 / Área não revestida = 9500.*

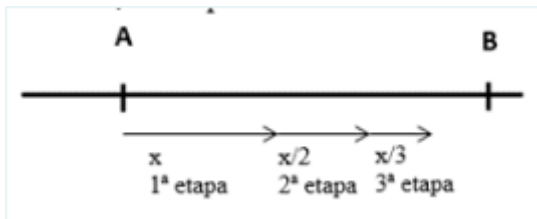
Resposta:

```
1  altura = int(input("Digite o valor da altura da parede: "))
2  comprimento = int(input("Digite o comprimento da parede: "))
3  lado = int(input("Digite o lado do azulejo: "))
4
5  qtdAzulejoAltura = altura // lado
6  qtdAzulejoComprimento = comprimento // lado
7  totalAzulejos = qtdAzulejoAltura * qtdAzulejoComprimento
8
9  areaRevestida = totalAzulejos * (lado * lado)
10 areaNaoRevestida = (altura * comprimento) - areaRevestida
11
12 print("A quantidade total de azulejos é: {}".format(totalAzulejos))
13 print("A área da parede não revestida é: {}".format(areaNaoRevestida))
```

31 – Um móvel desloca-se ao longo de uma reta determinada pelos pontos A e B, a partir do ponto A em direção ao ponto B. O movimento é realizado em várias etapas:

- na primeira etapa a distância percorrida é  $x$ ;
- na segunda etapa é  $x/2$ ;
- na terceira etapa é  $x/3$ ; e assim sucessivamente até que o móvel ultrapasse o ponto B.





"Conhecendo-se a distância entre os pontos A e B e a distância percorrida na primeira etapa do deslocamento, como determinar a quantidade de etapas necessárias para que o ponto B seja ultrapassado?"

Variáveis:

- Entrada: dab e x de tipo real.
- Saída: etap de tipo inteiro.

Exemplo da Tela de Execução:

Digite a distância ab: 10

Digite a distância x: 2

Número de etapas: 83

Resposta:

```
1  dab=float(input("Digite a distância ab:"))
2  x=float(input("digite a distância x:"))
3
4  n = 1
5
6  etapa = 0
7  distancia = 0
8
9  while distancia < dab:
10     distancia = distancia + x/n
11     n = n + 1
12     etapa = etapa + 1
13
14  print("numeros de etapas=",etapa)
```

32 - "Uma empresa decidiu dar uma gratificação de Natal aos seus funcionários baseada no número de horas extras e no número de horas que o funcionário faltou ao trabalho. O valor do prêmio é obtido pela consulta na tabela a seguir, em M é a quantidade de minutos equivalente à quantidade de horas obtida pela expressão:

$$M = (\text{quantidade de horas extras}) - \frac{2}{3} * (\text{quantidade de horas ausentes})$$

M(minutos)	Prêmio (R\$)
>2400	500
1800 até 2400	400
<=1800	100

Construir e utilizar uma função que tenha como parâmetros de entrada a quantidade de horas extras (HE) e a quantidade de horas que o funcionário faltou (HF) e como parâmetro de saída o valor do prêmio.

#### Exemplo da Tela de Execução:

digite a quantidade de horas extras: 20

digite a quantidade de horas que faltou: 4

O valor do prêmio é: 100

#### É obrigatório o uso da função def.

Usar CTRL C para copiar o código e a tela de execução e CTRL V para colar o código e a tela de execução. Usar na Tela de Execução 45 horas extras e 4 horas que faltou.

#### É proibido o uso de ALT PrtSc.

Resposta:

```
1  def Premios(A,B):
2      M = ((A)-2/3*(B))*60
3      if M >2400:
4          premio = 500
5      elif M > 1800 and M < 2400:
6          premio = 400
7      else:
8          premio = 100
9      return premio
10
11
12  HE = int(input("Digite as horas extras: "))
13  HA = int(input("Digite as horas ausentes: "))
14
15  P = Premios(HE,HA)
16  print(P)
17
18  #Digite as horas extras: 45
19  #Digite as horas ausentes: 4
20  #500
```

33 - No algoritmo a seguir todas as variáveis são do tipo inteiro. Simule sua execução supondo como entrada para n como valor 541. Marque no quadro de resposta a alternativa que contém a resposta para as variáveis a, soma e d.

```
leia n
a ← n div 100;
b ← (n mod 100) div 10;
c ← (n mod 100) mod 10;
inv ← c * 100 + b * 10 + a;
soma ← n + inv;
a ← (soma div 100);
b ← ((soma mod 100) div 10) * 2;
c ← ((soma mod 100) mod 10) * 3;
d ← (a + b + c) mod 10;
imprima d;
```

Resposta:

```
1  n = int(input("leia n: "))
2  a = n // 100
3  b = (n % 100) // 10
4  c = (n % 100) % 10
5  inv = (c * 100) + (b * 10) + a
6  soma = n + inv
7  a = soma // 100
8  b = ((soma % 100) // 10) * 2
9  c = ((soma % 100) % 10) * 3
10 d = (a + b + c) % 10
11 print("Valor de A:", a)
12 print("Valor da soma:", soma)
13 print("Valor de D:", d)
```

34 - Um órgão de defesa ao consumidor analisa um lote com várias dessas caixas. Se pelo menos 98% das caixas inspecionadas apresentarem quantidade efetiva maior do que 58 unidades, o lote é aprovado, caso contrário (porcentagem menor do que 98%) o lote é rejeitado. Conhecendo-se as quantidades efetivas das várias caixas inspecionadas, como determinar se o lote deve ser aprovado ou rejeitado? A 'saída' deverá ser uma das mensagens: lote aprovado ou lote rejeitado. Não é conhecida previamente a quantidade de caixas inspecionadas. Supor que após o último valor será anotado será digitado o valor zero (zero indica fim da entrada de dados).

*Exemplo da Tela de Execução:*

Entre quantidade 1: 60

Entre quantidade 2: 60

Entre quantidade 3: 59

Entre quantidade 4: 60

Entre quantidade 5: 60

Entre quantidade 6: 0

Porcentagem: 100.0 %

lote aprovado

Copie e cole o código e a tela de execução no espaço a seguir com os valores de entrada: 60;60;60;58;60;0."

Resposta:

```
1  qtd = 1
2  x = 1
3  t = 0
4  maior = 0
5  while (qtd > 0):
6      qtd = int(input("Entre quantidade %i: " % x))
7      x += 1
8      t += 1
9      if (qtd > 58):
10         maior += 1
11  Porcentagem = ((100 * maior) / (t - 1))
12  t -= 1
13  print("Porcentagem: %.2f%%" % Porcentagem)
14  if (maior >= ((98 * t) / 100)):
15      print("lote aprovado")
16  else:
17      print("lote reprovado")
18
```

35 - Considere que esse programa deve representar um método de resolução do seguinte problema:

A avaliação de mercado para um novo veículo é feita por uma revista, considerando-se três quesitos: preço final, custo de manutenção e aspectos de conforto.

Para cada quesito é obtida uma pontuação, de 0 até 10 com valores de 0.5 em 0.5, definida pelas respectivas equipes de avaliação.

A avaliação final é definida como média ponderada dessas três pontuações, atribuindo-se peso 2 para a menor das pontuações e peso 5 para cada uma das outras duas.

Conhecendo-se as pontuações dos três quesitos como obter o valor da avaliação final? Indique no quadro de respostas o preenchimento adequado de cada lacuna.

Resposta:

```
1 # Código incompleto:
2 pf = float(input("nota preço final:"))
3 cm = float(input("nota custo manutenção:"))
4 LACUNA 1 = float(input("nota aspecto de conforto:"))
5 menor = pf
6 if cm < menor:
7     menor = LACUNA 2
8 if LACUNA 3:
9     menor = ac
10 media = LACUNA 4
11 print("media = %.2f" % media)
12
13 # LACUNA 1: ac
14 # LACUNA 2: cm
15 # LACUNA 3: ac < menor
16 # LACUNA 4: (2*menor+5*(pf+cm+ac-menor))/12
```

36 –

Dado um número inteiro positivo  $n$ , calcular e exibir o valor da soma  $S$  dos  $n$  primeiros termos da série:

$$S = 2 + \frac{4}{3} + \frac{6}{5} + \frac{8}{7} + \dots$$

Exemplo:

- Para  $n = 3$ :

$$S = 2 + \frac{4}{3} + \frac{6}{5} = 4.53$$

- Para  $n = 4$ :

$$S = 2 + \frac{4}{3} + \frac{6}{5} + \frac{8}{7} = 5.68$$

O programa não pode permitir a entrada de dados inválidos, ou seja, um valor de  $n$  menor ou igual a zero.

Construir e utilizar uma função `soman` que tenha como parâmetro de entrada o número de termos da série (variável  $n$ ) e como parâmetro de saída o valor da soma  $S$ .

Resposta:

```
1 def soman(n):
2     x = 1
3     y = 1
4     s = 0
5     while (y <= n):
6         num = x + 1
7         den = x
8         s = s + (num / den)
9         x = x + 2
10        y += 1
11    return s
12
13    n = -1
14    while (n <= 0):
15        n = int(input("Digite o valor de n: "))
16
17    print("O valor da soma é: %.2f" % soman(n))
```

37 - Uma impressora tem capacidade para imprimir 5 páginas de texto por minuto. Conhecendo a quantidade de páginas de uma enciclopédia que essa impressora deve imprimir, como calcular o tempo necessário para realizar essa impressão?

Exemplo:

Entre com o número de páginas da enciclopédia: 2497

Tempo horas: 8

Tempo minutos: 19

Tempo segundos: 24

Resposta:

```
1  qpag = int(input("entre com o número de páginas da enciclopédia: "))
2  qm = qpag // 5
3  qs = int((((qpag % 5) / 5) * 60))
4  if (qm >= 60):
5      qh = qm // 60
6      qm = (qm % 60)
7  else:
8      qh = 0
9
10 print("tempo em horas: ", qh)
11 print("tempo em minutos: ", qm)
12 print("tempo em segundos: ", qs)
```

38 - Durante os 30 dias do mês de junho foram anotadas as quantidades diárias de ocorrências numa delegacia. Conhecendo-se a série de quantidades anotadas, como determinar a média diária de ocorrências registradas?

Resposta:

```
1  Soma = 0
2  for dia in range(1, 31):
3      qd = int(input("Digite a quantidade do dia: "))
4      Soma = Soma + qd
5  Média = Soma / 30
6  print("Média = %.2f" % Média)
```

39- Leia um valor inteiro. A seguir, calcule o menor número de notas possíveis (cédulas) no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2 e 1. A seguir mostre o **valor lido** e a relação de notas necessárias.

**Exemplo:**

**Entrada:**

Digite a quantia R\$ 576

**Saída:**

5 nota(s) de R\$ 100,00

1 nota(s) de R\$ 50,00

1 nota(s) de R\$ 20,00

0 nota(s) de R\$ 10,00

1 nota(s) de R\$ 5,00

0 nota(s) de R\$ 2,00

1 nota(s) de R\$ 1,00

Resposta:

```
1  n = (int((input("Digite a quantia: R$"))))
2
3  n100 = n //100
4  n100r = n % 100
5
6  n50 = n100r// 50
7  n50r = n100r % 50
8
9  n20 = n50r // 20
10 n20r = n50r % 20
11
12 n10 = n20r // 10
13 n10r = n20r % 10
14
15 n5 = n10r // 5
16 n5r = n10r % 5
17
18 n2 = n5r // 2
19 n2r = n5r % 2
20
21 n1 = n2r // 1
22 n1r = n2r % 1
23
24 print("Quantidade de notas de R$100,00: ", n100)
25 print("Quantidade de notas de R$50,00: ", n50)
26 print("Quantidade de notas de R$20,00: ", n20)
27 print("Quantidade de notas de R$10,00: ", n10)
28 print("Quantidade de notas de R$5,00: ", n5)
29 print("Quantidade de notas de R$2,00: ", n2)
30 print("Quantidade de notas de R$1,00: ", n1)
```

40 - Faça um programa que peça as quatro notas (entre 0 e 10) de 10 alunos, calcule e armazene num vetor a média de cada aluno, imprima o número de alunos com média maior ou igual a 7,0.

- Utilizar uma função **entrada**, para a entrada das notas e cálculo da média.
- Utilizar uma função para contar o número de aprovados.
- Utilizar uma função **saída** para exibir a média de cada aluno e o número de alunos aprovados.

Resposta:

```
1  def entrada():
2      media = []
3      print("Digite 4 notas:")
4      for i in range(9):
5          acum = 0
6          print("Notas do aluno %d " % (i+1))
7          for n in range(4):
8              n = float(input())
9              while n < 0 or n > 10:
10                 print("Nota fora do intervalo, digite novamente")
11                 n = float(input())
12                 acum = acum + n
13             m = acum / 4
14             media.append(m)
15         return (media)
16
17 def aprovados(media):
18     ct = 0
19     for i in range(9):
20         if media[i] >= 7.0:
21             ct = ct + 1
22     return ct
23
24 def saida(ap, media):
25     print(media)
26     print("Qtde aprovados: ", ap)
27
28 # principal
29 md = entrada()
30 ap = aprovados(md)
31 saida(ap, md)
```



41 - A média aritmética ponderada  $X_p$  de um conjunto de números  $x_1, x_2, x_3, \dots, x_n$  cuja importância relativa ("peso") é respectivamente  $p_1, p_2, p_3, \dots, p_n$  é calculada da seguinte maneira:

$$x_p = \frac{p_1 x_1 + p_2 x_2 + p_3 x_3 + \dots + p_n x_n}{p_1 + p_2 + p_3 + \dots + p_n}$$

Como obter a média aritmética ponderada de, no máximo, 50 números, conhecendo-se a quantidade de números que serão considerados, os números e seus respectivos pesos?

O código abaixo dá a solução deste problema, mas está incompleto.

- A quantidade de números que serão considerados é armazenada na variável  $n$ , os números são guardados na lista  $N$  de 50 componentes e os pesos de cada número são guardados na lista  $P$  de 50 componentes.
- Na variável auxiliar  $numerador$  é armazenado o cálculo do numerador da média aritmética ponderada conforme definição.
- Na variável auxiliar  $den$  é armazenado o resultado do denominador da média aritmética ponderada conforme definição.
- A média ponderada é armazenada na variável  $zp$ .

**Complete as lacunas de 1 a 4.**

**Principal**

Entrada:

$N$  e  $P$  tipo float;  $n$  tipo int

Saída:

$zp$  tipo float

Resposta:

```
1  N = []
2  P = []
3  n = int(input("Insira quantos números serão considerados: "))
4  den = 0
5  numerador = 0
6
7  for i in range(0, n, 1):
8      N.append(float(input("Digite o número %d: " % (i + 1))))
9      P.append(float(input("Digite o peso %d: " % (i + 1))))
10
11  for j in range(0, n, 1):
12      numerador = numerador + N[j] * P[j]
13
14  for k in range(0, n, 1):
15      den = den + P[k]
16
17  xp = numerador / den
18  print("A média ponderada é %.2f" % xp)
```

42 - Obter o espelho de um valor inteiro positivo conhecido.

Observação: entenda-se espelho como o valor obtido pela leitura invertida do valor.

Exemplos:

- O espelho de 3629 é 9263
- O espelho de 301 é 103
- O espelho de 2000 é 2
- O espelho de 5 é 5

Resposta:

```
1  valor = int(input("Digite o valor:"))
2  r = 0
3  esp = 0
4
5  while valor > 0:
6      r = valor % 10
7      esp = (esp * 10) + r
8      valor = valor // 10
9
10 print("Resultado:", esp)
```

43 – Verificar se um número é par ou ímpar.

Resposta:

```
1 n = int(input("Digite um número inteiro positivo:"))
2 if n % 2 != 0:
3     print("número ímpar")
4 else:
5     print("número par")
```

44- Considere o seguinte problema: Dado um número inteiro positivo n, calcular e exibir o valor da soma S:  $S = 1/n + (2/n-1) + (3/n-2) + \dots + n/1$  O programa não pode permitir a entrada de dados inválidos, ou seja, um valor de n menor ou igual a zero.

Resposta:

```
1 n = -1
2 while n <= 0:
3     n = int(input("Digite o número de parcelas: "))
4 soma = 0
5 den = n
6
7 for k in range(1, n + 1, 1): # SE VOCÊ QUISER REPETIR UM BLOCO 20 VEZES VOCÊ
8     soma = soma + k / den    # # PODE ALTERAR AQUI
9     den = den - 1
10 print("soma = %.2f" % soma)
```

45 – Números inteiro Positivos de uma série for:

```
1 cont = 0
2 quant = int(input("Entre com a quantidade de número: "))
3 for i in range(1, quant + 1):
4     n = int(input("Entre um número positivo: "))
5     if n % 2 == 1:
6         cont = cont + 1
7
8 p = (cont / quant) * 100
9 print("porcentagem de números ímpares = %.2f" % p, "%")
```

46 - Crie um programa que receba uma sequência de números inteiros positivos e calcule o produtório (multiplicação) de todos os números pares inseridos. O programa deve atender às seguintes condições:

- O programa encerra a execução quando o número **0** for digitado.
- Caso o número digitado seja negativo, o programa deve solicitar novamente um número válido (inteiro e positivo).
- O primeiro número digitado pode ser **0**, e, nesse caso, o produtório será **0**.

- Se nenhum número par for inserido antes de digitar **0**, o programa deve retornar o resultado **0**.
- O programa deve exibir o resultado do produtório dos números pares ao final da execução.

#### Exemplo de saída esperada:

- Entrada: 4, 6, 3, 0  
Saída: "O produtório dos números pares é: 24"
- Entrada: 0  
Saída: "O produtório dos números pares é: 0"

Resposta:

```
n = -1 # Valor aleatório de "n" para entrar no WHILE
p = 0 # Valor inicial do produtório consecutivo
k = 0 # Contador para caso o primeiro número digitado seja "0"

while n != 0: # Verificação de "n" para encerrar a simulação
    n = int(input("Digite o valor inteiro e positivo: "))
    while n < 0: # Verificação se o valor é válido, se ele é inteiro e positivo
        print("Valor incorreto, apresente valor inteiro e positivo")
        n = int(input("Digite o valor inteiro e positivo: "))

    if n % 2 == 0 and n != 0: # Se "n" for par e diferente de "0", entra no IF
        if p == 0: # Verifica se "p" ainda é zero
            p = n # Se "p" for zero, "p" recebe o valor de "n" para dar procedência às próximas operações
        elif p != 0: # Se "p" já tiver um valor, basta apenas multiplicá-lo pelo novo "n" e jogar esse novo valor de volta ao "p"
            p = p * n

    if n == 0 and k == 0: # Se de início colocar-se "0", então "p" não deverá ter nenhum valor.
        p = 0 # Ambos "n" e "k" são comparados para saber se o primeiro valor da simulação foi "0", e "k" também será "0" devido ao contador.
    k = k + 1 # Somente nesse ponto "k" atribui outros valores para sair das condições que podiam implicar nos resultados

print("O produtório dos números pares é: ", p)
```

47 - TEMPERATURA (imprimir a maior, a menor e a média da temperatura):

Resposta:

```
1 temp = [-10, -8, 0, 1.2, 5, -2, -4]
2 maior = temp[0]
3 menor = temp[0]
4 soma = 0
5
6 for i in range(0, len(temp), 1):
7     if temp[i] >= maior:
8         maior = temp[i]
9     if temp[i] <= menor:
10        menor = temp[i]
11        soma = soma + temp[i]
12
13 print("Maior temperatura: ", maior)
14 print("Menor temperatura: ", menor)
15 print("A média das temperaturas: %.2f" % (soma / len(temp)))
```

48 - "Conhecendo-se uma sequência de 10 valores correspondentes aos preços de um produto em 10 lojas varejistas, como determinar em quantas das lojas o preço do produto é superior ao preço médio dessas 10 lojas?"

Resposta:

```
1  prec = []
2  soma = 0
3  for i in range(0, 10, 1):
4      p = float(input("Entre com o preço da Loja %d: " % (i+1)))
5      prec.append(p) # quando ele coloca append ele guarda o que a lista quer
6      soma = soma + prec[i]
7
8  pm = soma / 10
9  q = 0
10
11 for i in range(0, 10, 1):
12     if prec[i] > pm:
13         q += 1
14
15 print("Quantidade de lojas com preços acima da média:", q)
```

49 – Raízes reais de uma equação

Resposta:

```
1  from math import *
2
3  a = int(input("Termo a: "))
4  b = int(input("Termo b: "))
5  c = int(input("Termo c: "))
6
7  if a != 0:
8      delta = b**2 - (4 * a * c)
9      if delta > 0:
10         x1 = (-b + sqrt(delta)) / (2 * a)
11         x2 = (-b - sqrt(delta)) / (2 * a)
12         print("%.2f e %.2f" % (x1, x2))
13     elif delta == 0:
14         x1 = -b / (2 * a)
15         print("%.2f" % x1)
16     else:
17         print("A equação não tem raízes reais")
18 else:
19     print("Não é uma equação do segundo grau")
```

## 50 – Número de Divisores

Resposta:

```
1  def entrada():
2      n = -1
3      while n <= 0:
4          n = int(input("Digite um número inteiro positivo: "))
5      return n
6
7  def contdiv(a):
8      ndiv = 0
9      for cont in range(1, a + 1):
10         resto = a % cont
11         if resto == 0:
12             ndiv = ndiv + 1
13     return ndiv
14
15 x = entrada()
16 y = contdiv(x)
17 print("%d tem %d divisores" % (x, y))
```

## 51 – Multiplicação sem Uso de \*:

Resposta:

```
1  a = int(input("Digite o primeiro número inteiro: "))
2  b = int(input("Digite o segundo número inteiro: "))
3
4  soma = 0
5  cont = 1
6  while cont <= a:
7      soma = soma + b
8      cont = cont + 1
9
10 print("O produto de %d x %d é = %d" % (a, b, soma))
```

## 52 – Notas: média

Resposta:

```
1  def ordemcrescente(A, B, C):
2      if A > B:
3          aux = A
4          A = B
5          B = aux
6      if A > C:
7          aux = A
8          A = C
9          C = aux
10     if B > C:
11         aux = B
12         B = C
13         C = aux
14     return A, B, C
15
16 N1 = float(input("Entre com a nota 1: "))
17 N2 = float(input("Entre com a nota 2: "))
18 N3 = float(input("Entre com a nota 3: "))
19
20 X, Y, Z = ordemcrescente(N1, N2, N3)
21 mf = (3 * X + 7 * Z) / 10
22
23 print("Média Final: %.2f" % mf)
```

53 – Algoritmo:

Resposta:

```
1  n = int(input("Digite um valor para n: "))
2  a = n // 100
3  b = (n % 100) // 10
4  c = (n % 100) % 10
5  inv = c * 100 + b * 10 + a
6  soma = n + inv
7  a = (soma // 100)
8  b = ((soma % 100) // 10) * 2
9  c = ((soma % 100) % 10) * 3
10 d = (a + b + c) % 10
11
12 print("a:", a)
13 print("b:", b)
14 print("c:", c)
15 print("d:", d)
```

54 – Múltiplos de 5:

Resposta:

```
1  a = int(input("Digite um número para a: "))
2  b = int(input("Digite um número para b: "))
3  if a > b:
4      print("Números Incorretos")
5  else:
6      soma = a + b
7      multiplos = 0
8      if a % 5 == 0:
9          multiplos += 1
10     if b % 5 == 0:
11         multiplos += 1
12     while b > a + 1:
13         b -= 1
14         soma = soma + b
15         if soma % 5 == 0:
16             multiplos += 1
17
18     print("soma dos números inteiros: %d" % soma)
19     print("quantidade de números múltiplos de 5: %d" % multiplos)
```

55 -Reservatório de Água (Cálculo de tempo através de capacidade e vazão):

Resposta:

```
1  C = int(input("Digite a capacidade do reservatório: "))
2  V = int(input("Digite a vazão da bomba: "))
3
4  Ts = C // V
5  Qh = Ts // 3600
6  R = Ts % 3600
7  Qm = R // 60
8  Qs = R % 60
9
10 print("Horas para abastecer o reservatório:", Qh)
11 print("Minutos para abastecer o reservatório:", Qm)
12 print("Segundos para abastecer o reservatório:", Qs)
```

56 - Soma de Algarismos:

Resposta:

```
1  número = int(input("Digite um número com 3 algarismos: "))
2  pn = número // 100
3  resto = número % 100
4  sn = resto // 10
5  tn = resto % 10
6  soma = pn + sn + tn
7  print("Soma dos algarismos:", soma)
```

57 – Cálculo de IMC:

Resposta:

```
1  peso = []
2  altura = []
3  IMC = []
4
5  for i in range(0, 5, 1):
6      peso.append(float(input("Digite o peso da pessoa %d: " % (i + 1))))
7      altura.append(float(input("Digite a altura da pessoa %d: " % (i + 1))))
8
9  for j in range(0, 5, 1):
10     IMC.append(peso[j] / (altura[j] ** 2))
11
12  for k in range(0, 5, 1):
13     print("peso =", peso[k], "altura =", altura[k], "IMC=%.2f" % IMC[k])
```

58 - Considere o seguinte problema:

"Dado o valor de  $n$  ( $n \geq 1$  e  $n \leq 60$ ), como obter o valor da soma  $S$  da série a seguir:

$$S = \frac{60}{6} + \frac{59}{12} + \frac{58}{18} + \dots + \frac{1}{6 \times n}$$

Resposta:

```
1  n = int(input("Digite o valor de n: "))
2  d = 6
3  s = 0
4  num = 60.0
5  for i in range(1, n + 1, 1):
6      m = num / d
7      s = s + m
8      d = d + 6
9      num = num - 1
10  print("soma=%.2f" % s)
```



59 - Calcule o consumo médio de um automóvel sendo fornecidos a distância total percorrida (em Km) e o total de combustível gasto (em litros).

Entrada:

O arquivo de entrada contém dois valores: um valor inteiro XXX representando a distância total percorrida (em Km), e um valor real YYY representando o total de combustível gasto, com um dígito após o ponto decimal.

Saída:

Apresente o valor que representa o consumo médio do automóvel com 3 casas após a vírgula, seguido da mensagem "km/l".

Resposta:

```
1 x = int(input("Digite a distância total percorrida(Km): "))
2 y = float(input("Digite o total de combustível gasto(L): "))
3
4 cm = x / y
5
6 print("Consumo médio do automóvel(km/L): %.3f" % cm)
```

60 - "Calcular e exibir a média final de um aluno, a partir das 3 avaliações realizadas. À menor nota é atribuído peso 3 e à maior nota é atribuído peso 7 (a nota com valor intermediário é descartada).

Construir e utilizar a função auxiliar MaiorMenor que tenha como parâmetros de entrada as três notas - a, b, e c de tipo float, e como parâmetros de saída a maior e a menor das notas – max e min de tipo float."

Resposta:

```
1 def MaiorMenor(a, b, c):
2     notas = []
3     maior = max(a, b, c)
4     menor = min(a, b, c)
5     notas.append(menor)
6     notas.append(maior)
7     menorNota = notas[0] * 0.3
8     maiorNota = notas[1] * 0.7
9     print("Maior Nota foi de %.2f, com peso 7 aplicado temos %.2f" % (notas[1], maiorNota))
10    print("Menor Nota foi de %.2f, com peso 3 aplicado temos %.2f" % (notas[0], menorNota))
11
12 a = float(input("Digite uma nota: "))
13 b = float(input("Digite uma nota: "))
14 c = float(input("Digite uma nota: "))
15 MaiorMenor(a, b, c)
```

61- "A equipe de engenharia responsável pela programação da produção da petroquímica SoftShell calculou os componentes dos custos de produção de um determinado solvente. O custo fixo calculado resultou R\$ 80.000,00 (o custo fixo, em geral, é composto por: aquisição de máquinas/equipamentos, contratação/treinamento de pessoal, alocação física do dispositivo de produção, ...). O custo variável resultou R\$ 6,00 por litro do solvente produzido, ou seja: a produção de cada litro custa R\$ 6,00 para a empresa, além da parcela denominada custo fixo (o custo variável, em geral, é composto por matéria-prima, energia, impostos, embalagens, refugos, ...). Sabendo-se que cada litro será vendido a R\$ 10,00 e conhecendo-se o valor do lucro que a empresa deseja obter, como determinar a quantidade de litros do solvente que deve ser produzida? Suponha que todo o volume produzido será vendido."

Resposta:

```
1  lucroi = float(input('Digite o lucro que gostaria: R$ '))
2  lminimo = 20000
3  litrosf = lminimo + lucroi / 4
4  print("Litros necessários:", litrosf)
```

62 - "Faça um programa que leia um vetor X[10]. Substitua, a seguir, todos os valores nulos ou negativos do vetor por 1. Em seguida, mostre o vetor XXX.

**Entrada:**

A entrada contém 10 valores inteiros, podendo ser positivos ou negativos.

**Saída:**

Para cada posição do vetor, escreva 'X[i] = c', onde i é a posição do vetor XXX e c é o valor armazenado naquela posição."

Resposta:

```
1  lista = []
2  for i in range(10):
3      x = int(input(""))
4      if x <= 0:
5          x = 1
6          lista.append(x)
7      else:
8          lista.append(x)
9
10 print("X[0] = ", lista[0])
11 print("X[1] = ", lista[1])
12 print("X[2] = ", lista[2])
13 print("X[3] = ", lista[3])
14 print("X[4] = ", lista[4])
15 print("X[5] = ", lista[5])
16 print("X[6] = ", lista[6])
17 print("X[7] = ", lista[7])
18 print("X[8] = ", lista[8])
19 print("X[9] = ", lista[9])
```

63 - "Faça um programa que solicite do usuário uma quantidade de dias, horas, minutos e segundos. Calcule e imprima o total convertido em somente segundos."

Resposta:

```
1 d = int(input("Digite a quantidade de dias: "))
2 h = int(input("Digite a quantidade de horas: "))
3 m = int(input("Digite a quantidade de minutos: "))
4 s = int(input("Digite a quantidade de segundos: "))
5
6 horast = h + (d * 24)
7 mintotal = m + (horast * 60)
8 sectotal = s + (mintotal * 60)
9
10 print("Total de segundos:", sectotal)
```

64 - "Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$ 60,00 por dia e R\$ 0,15 por km rodado."

Resposta:

```
1 km = float(input("Digite a quantidade de kms percorridos: "))
2 dias = int(input("Digite a quantidade de dias que o carro foi alugado: "))
3
4 preco = (dias * 60) + (km * 0.15)
5
6 print("Preço final:", preco)
```

65 - "Faça um programa, utilizando funções, que receba três números inteiros e positivos e que forneça a soma desses três números.

Para este exercício crie três funções:

- entrada(): retorna um número digitado (fazer a verificação se é positivo);
- calculaSoma(a, b, c): recebe 3 números inteiros e positivos e retorna a soma deles;
- principal: chamada das funções criadas (chama 3 vezes a entrada, sendo uma para cada número e a função para somar) e depois mostre o resultado."

Resposta:

```

1  # Função que retorna o número digitado
2  def entrada():
3      x = int(input("Digite o número: "))
4      while x <= 0:
5          x = int(input("Digite o número: "))
6      return x
7
8  # Função que faz a soma de 3 números positivos
9  def calculaSoma(a, b, c):
10     d = a + b + c
11     return d
12
13 # Principal
14 n1 = entrada()
15 n2 = entrada()
16 n3 = entrada()
17 s = calculaSoma(n1, n2, n3) # Linha de chamada
18 print("Soma =", s)

```

66 - Faça um Programa para leitura de três notas parciais de um aluno. O programa deve calcular a média  $M = (N1 + N2*2 + N3*3)/6$  alcançada pelo aluno e apresentar:

- A mensagem "**Aprovado**", se a média for maior ou igual a 5, com a respectiva média alcançada;
- A mensagem "**Reprovado**", se a média for menor do que 5, com a respectiva média alcançada.

Resposta:

```

1  x = float(input("Digite a 1ª nota: "))
2  y = float(input("Digite a 2ª nota: "))
3  z = float(input("Digite a 3ª nota: "))
4
5  media = (x + y*2 + z*3) / 6
6
7  if media >= 5:
8      print("Aprovado")
9  else:
10     print("Reprovado")

```

67 - Desenvolva um programa que receba a quantidade de alunos de uma turma e as médias de cada aluno. O programa deve verificar se o número de alunos é válido (maior que 0) e calcular a média da turma. Para cada aluno, exiba uma mensagem indicando se ele foi "**APROVADO**" (média maior ou igual a 6) ou "**REPROVADO**" (média menor que 6). Por fim, mostre a média geral da turma formatada com duas casas decimais.

**Regras:**

- Se a quantidade de alunos for menor ou igual a zero, exiba a mensagem "**NÃO HOUVE PROCESSAMENTO**" e solicite novamente o número de alunos.
- Para cada aluno, insira sua média.
- Exiba uma mensagem individual de aprovação ou reprovação para cada média informada.
- Calcule e mostre a média geral da turma.

Resposta:

```

1  n = int(input("Digite a quantidade de alunos: "))
2  k = 0
3  while n <= 0:
4      print("NÃO HOUVE PROCESSAMENTO")
5      n = int(input("Digite a quantidade de alunos: "))
6
7  for i in range(1, n + 1, 1):
8      m = float(input("Digite a média: "))
9      k = k + m
10     if m >= 6:
11         print("PARABÉNS VOCÊ ESTÁ APROVADO")
12     elif m < 6:
13         print("VOCÊ ESTÁ REPROVADO")
14
15     media = k / n
16     print("Média da turma: %.2f" % media)

```

68 - Escreva um programa que leia números digitados pelo usuário. O programa deve ler os números até que **0 (zero)** seja digitado. Quando o número **0** for digitado, o programa deve exibir:

- A quantidade de números que foram digitados;
- A somatória destes números;
- A média aritmética.

Resposta:

```

1  soma = 0
2  k = 0
3
4  while True:
5      n = float(input("Digite um número: "))
6      if n == 0:
7          break
8      soma = soma + n
9      k = k + 1
10
11     media = soma / k
12
13     print("Média: %.2f" % media)
14     print("Soma:", soma)
15     print("Quantidade de números:", k)
16

```

69 - Conhecendo-se a base e a altura de um retângulo, calcular e exibir a área, perímetro e a diagonal. Construir e utilizar uma função que tenha como parâmetros de entrada a base e a altura e como parâmetros de entrada/saída a área e o perímetro. Construir e utilizar outra função que tenha como parâmetros de entrada a base e a altura e como parâmetro de saída a diagonal.

Resposta:

```
1  from math import *
2
3  def contas(base, altura):
4      area = base * altura
5      perimetro = 2 * base + 2 * altura
6      diagonal = sqrt(base**2 + altura**2)
7      print("Área:", area)
8      print("Perímetro:", perimetro)
9      print("Diagonal:", diagonal)
10
11  base = int(input("Digite a base do retângulo: "))
12  altura = int(input("Digite a altura do retângulo: "))
13  contas(base, altura)
```

70 – Função Ordem Crescente:

Resposta:

```
1  def ordemdecrescente(a, b):
2      if a > b:
3          t = a
4          h = b
5      else:
6          t = b
7          h = a
8      return t, h
9
10 def diferenca(c, d):
11     j = c - d
12     return j
13
14 n1 = float(input("Digite o primeiro número: "))
15 n2 = float(input("Digite o segundo número: "))
16 maior, menor = ordemdecrescente(n1, n2)
17 print("Ordem decrescente:", (maior, menor))
18 diferenca = diferenca(maior, menor)
19 print("A diferença é: %.2f" % diferenca)
```

71 - Elabore um programa que mostre o preço de etiqueta de um produto e calcule e mostre o quanto deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento.

Utilize os códigos da tabela seguinte para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

Código	Condições de pagamento
1	À vista em dinheiro ou cheque, recebe 10% de desconto
2	À vista no cartão de crédito, recebe 5% de desconto
3	Em 2 vezes, preço normal de etiqueta sem juros
4	Em 3 vezes, preço normal de etiqueta mais juros de 10%

Resposta:

```
1 p = float(input("Digite o preço do produto: "))
2 c = int(input("Digite o código de pagamento: "))
3
4 if c == 1:
5     valor = p * 0.9
6 if c == 2:
7     valor = p * 0.95
8 if c == 3:
9     valor = p
10 if c == 4:
11     valor = p * 1.1
12
13 print("O valor final é:", valor)
```

72 - Conhecendo-se uma sequência de 10 valores correspondentes aos preços de um produto em 10 lojas varejistas, como determinar em quantas das lojas o preço do produto é superior ao preço médio dessas 10 lojas?

Resposta:

```
1 prec = []
2 soma = 0
3
4 for i in range(0, 10, 1):
5     p = float(input("Entre com o preço da loja %d: " % (i + 1)))
6     prec.append(p) # Adiciona o valor à lista 'prec'
7     soma = soma + prec[i]
8
9 pm = soma / 10
10 q = 0
11
12 for i in range(0, 10, 1):
13     if prec[i] > pm:
14         q = q + 1
15
16 print("Quantidade de lojas com preços acima da média:", q)
17
```

73 - Considere o seguinte problema:

"Conhecendo-se o valor total do pedido e o número de produtos adquiridos, calcular e exibir o valor a ser pago considerando as seguintes condições:

- Pedidos acima de R\$ 200,00 com um único produto recebem desconto de 8% e frete grátis;
- Pedidos acima de R\$ 200,00 com mais de um produto, recebem desconto de 8% e pagam frete de R\$ 7,00;
- Pedidos que custam entre R\$ 15,00 e R\$ 200,00 (inclusive) recebem desconto de 7% e pagam frete de R\$ 15,00;
- Demais situações não recebem desconto e pagam frete de R\$ 15,00."

Resposta:

```
1  x = int(input("Digite o número de produtos: "))
2  y = float(input("Digite o valor total do pedido: "))
3
4  if y > 200 and x == 1:
5      desconto = 0.08
6      frete = 0
7  elif y > 200 and x > 1:
8      desconto = 0.08
9      frete = 7
10 elif y >= 15 and y <= 200:
11     desconto = 0.07
12     frete = 15
13 else:
14     desconto = 0
15     frete = 15
16
17 desconto = y * desconto
18 valor = y - desconto + frete
19
20 print("O valor a ser pago é:", valor)
```

74 - Escreva um programa que, dado um número inteiro positivo, calcular e exibir a quantidade de números pares até o número digitado.

Resposta:

```
1  x = int(input("Digite um número inteiro: "))
2  count = 0
3
4  while x > 0:
5      if x % 2 == 0:
6          x -= 2
7          count += 1
8      else:
9          x -= 1
10
11 print("Quantidade de números pares:", count)
```



75 - Escreva uma função que retorne o maior de dois números. A função deve se chamar máximo (x, y).

Resposta:

```
1 def maximo(x, y):
2     if x > y:
3         print("x:", x)
4     else:
5         print("y:", y)
6
7     x = int(input("Digite x: "))
8     y = int(input("Digite y: "))
9     maximo(x, y)
```

76 - Escreva uma função chamada múltiplo(x, y) que receba dois números e retorne **True** se o primeiro for múltiplo do segundo número.

Resposta:

```
1 def multiplo(x, y):
2     if x % y == 0:
3         print("True")
4     else:
5         print("Error")
6
7     x = int(input("Digite um número: "))
8     y = int(input("Digite outro número: "))
9     multiplo(x, y)
```

77 - Escreva uma função que receba a base e a altura de um triângulo e retorne sua área  $A = \text{base} \times \text{altura} / 2$

Resposta:

```
1 def a(base, altura):
2     a = base * altura / 2
3     print(a)
4
5     base = int(input("Digite o valor da base: "))
6     altura = int(input("Digite o valor da altura: "))
7     a(base, altura)
8
```

78 - Considere o seguinte algoritmo:

algoritmoQ1( )

```
imprima "valor x?"; leia x;
y ← 1; w ← 0; c ← 1;
enquanto (c ≤ x) faça
    t ← c div y;
    y ← y + 1;
    q ← c mod y;
    w ← w + t + q;
    c ← c + 4;
imprima w;
```

Simule a execução do algoritmo acima, considere para a entrada no valor de x o **primeiro algarismo do número de matrícula** ou o **dígito do número de matrícula**. Indique a resposta para o valor final das variáveis w e c.

Resposta: 10, 13

```
1  x = int(input("Digite o valor de x: "))
2  y = 1
3  w = 0
4  c = 1
5
6  while (c <= x):
7      t = c // y
8      y = y + 1
9      q = c % y
10     w = w + t + q
11     c = c + 4
12
13     print("Valor de w:", w)
14     print("Valor de c:", c)
15
```

79 - Faça um programa que leia três números e mostre o maior e o menor deles.

Resposta:

```
1  maior = 0
2  menor = 0
3
4  n1 = int(input("Digite o primeiro número: "))
5  n2 = int(input("Digite o segundo número: "))
6  n3 = int(input("Digite o terceiro número: "))
7
8  if n1 >= n2 and n1 >= n3:
9      maior = n1
10
11  if n2 >= n1 and n2 >= n3:
12      maior = n2
13
14  if n3 >= n1 and n3 >= n2:
15      maior = n3
16
17  if n1 <= n2 and n1 <= n3:
18      menor = n1
19
20  if n2 <= n1 and n2 <= n3:
21      menor = n2
22
23  if n3 <= n1 and n3 <= n2:
24      menor = n3
25
26  print("O maior número é: {} e o menor número é: {}".format(maior, menor))
27
```

80 - Faça um programa que peça dois números e imprima o maior deles. Caso forem iguais, o programa também deverá apontar.

Resposta:

```

1 maior = 0
2 a = int(input("Digite o primeiro número inteiro: "))
3 b = int(input("Digite o segundo número inteiro: "))
4
5 if a >= b:
6     maior = a
7 if a <= b:
8     maior = b
9 if a == b:
10     print("Os números são iguais")
11
12 print("O maior número é: {}".format(maior))
13

```

Operadores	Significado	Exemplo	Resultado
$x + y$	soma	$2 + 4$	6
$x - y$	subtração	$2 - 4$	-2
$x * y$	multiplicação	$2 * 4$	8
$x / y$	divisão	$25 / 10$	2.5
$x // y$	divisão truncada	$25 // 10$	2
		$25 // 10.0$	2.0
$x \% y$	resto da divisão	$25 \% 10$	5
$x ** y$	potenciação	$2 ** 4$	16

*from math import \**

Sintaxe	Descrição
$\cos(x)$	retorna o cosseno de x
$\exp(x)$	retorna o exponencial de x ( $e^x$ )
$\sin(x)$	retorna o seno de x
$\text{fabs}(x)$	retorne o valor absoluto de x
$\text{factorial}(x)$	retorna $x!$
$\log(x,b)$	retorna $\log_b x$ (se b for omitido, retorna $\log x$ na base e)
$\log10(x)$	retorna $\log_{10} x$
$\pi$	constante $\pi$ (aproximadamente 3.1415926535897931)
$\text{pow}(x,y)$	retorna $x^y$
$\text{sqrt}(x)$	retorna a raiz quadrada de x
$\text{round}(x,n)$	número x arredondado para n dígitos

Operadores		Exempl o	Resultad o
==	igual a	10 == 10	True
!=	diferente	10 != 10	False
<	menor que	10 < 20	True
>	maior que	4 > 7	False
<=	menor ou igual a	10 <= 10	True
>=	maior ou igual a	5 >= 7	False

marcador	tipo
%d	Número inteiro (int)
%f	Número decimais(float)
%s	strings

Operação	Algoritmo	Phyton
/ (divisão truncada)	x div y	x//y
/ (divisão normal)	x/y	x/y
% (resto da divisão inteira)	x mod y	x%y

- Menor que: <
- Menor ou igual: <=
- Maior que: >
- Maior ou igual: >=
- Igualdade: ==
- Diferença: !=

```
se = if
senão se = elif
senão = else
```

mod = %

Operador	Descrição	Abreviação (forma composta)	Exemplo	Resultado
<code>+</code>	Soma	<code>+=</code>	<code>x += 5</code>	<code>x = x + 5</code>
<code>-</code>	Subtração	<code>-=</code>	<code>x -= 3</code>	<code>x = x - 3</code>
<code>*</code>	Multiplicação	<code>*=</code>	<code>x *= 2</code>	<code>x = x * 2</code>
<code>/</code>	Divisão	<code>/=</code>	<code>x /= 4</code>	<code>x = x / 4</code>
<code>%</code>	Módulo (resto da divisão)	<code>%=</code>	<code>x %= 3</code>	<code>x = x % 3</code>
<code>**</code>	Exponenciação (potência)	<code>**=</code>	<code>x **= 2</code>	<code>x = x ** 2</code>
<code>//</code>	Divisão inteira (sem casas dec.)	<code>//=</code>	<code>x //= 3</code>	<code>x = x // 3</code>

Operador	Descrição	Exemplo	Resultado
<code>==</code>	Igual a	<code>x == y</code>	<code>True</code> ou <code>False</code>
<code>!=</code>	Diferente de	<code>x != y</code>	<code>True</code> ou <code>False</code>
<code>&gt;</code>	Maior que	<code>x &gt; y</code>	<code>True</code> ou <code>False</code>
<code>&lt;</code>	Menor que	<code>x &lt; y</code>	<code>True</code> ou <code>False</code>
<code>&gt;=</code>	Maior ou igual	<code>x &gt;= y</code>	<code>True</code> ou <code>False</code>
<code>&lt;=</code>	Menor ou igual	<code>x &lt;= y</code>	<code>True</code> ou <code>False</code>

Operador	Descrição	Exemplo	Resultado
<code>and</code>	E (conjunção)	<code>x &gt; 0 and y &lt; 10</code>	<code>True</code> se ambos forem verdadeiros
<code>or</code>	Ou (disjunção)	<code>x &gt; 0 or y &lt; 10</code>	<code>True</code> se pelo menos um for verdadeiro
<code>not</code>	Negação lógica	<code>not (x &gt; 0)</code>	Inverte o valor lógico ( <code>True</code> → <code>False</code> )

Operador	Descrição	Exemplo	Resultado
<code>in</code>	Verifica se um elemento está em algo	<code>'a' in 'abc'</code>	<code>True</code>
<code>not in</code>	Verifica se um elemento não está em algo	<code>'d' not in 'abc'</code>	<code>True</code>
<code>is</code>	Verifica se duas variáveis apontam para o mesmo objeto	<code>x is y</code>	<code>True</code> ou <code>False</code>
<code>is not</code>	Verifica se duas variáveis não apontam para o mesmo objeto	<code>x is not y</code>	<code>True</code> ou <code>False</code>



