



Atividade – Teste de Mesa

01. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```

1 # Define uma função de nome "bem_vindo"
2 def bem_vindo():
3     # Instrução a ser executada quando a função "bem_vindo" for chamada.
4     print('Bem-vindo ao Python.')
5 # Define a função "mensagem", que recebe um argumento "msg".
6 def mensagem(msg):
7     # Instrução a ser executada quando a função for solicitada.
8     print(msg)
9 # Traz a função à tona, imprimindo a instrução "print('Bem-vindo ao Python.')".
10 bem_vindo()
11 # Traz a função à tona, e revela o conteúdo do argumento "msg".
12 mensagem("Curso de Programação Estruturada")

```

Preencha o teste de mesa para o código acima.

Memória		Tela
[bem_vindo]	[mensagem]	(10) Bem-vindo ao Python.
(10) Bem-vindo ao Python.	msg	(12) Curso de Programação Estruturada.
	(12) Curso de Programação Estruturada	

Descreva, resumidamente, o que faz o código: Define funções e as chamam para imprimir as mensagens

"Bem-vindo ao Python." e "Curso de Programação Estruturada".

02. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```

1 # Define a função "eh_par" e lhe concede o parâmetro "numero".
2 def eh_par(numero):
3     # Analisa se o resto da divisão da variável por 2 é igual a zero.
4     return numero % 2 == 0
5
6 # Retorna True para 2 pois o resto é igual a 0. Ou seja, a afirmação está correta.
7 print('2 é par?', eh_par(2))
8 # Retorna False para 3 pois o resto é diferente de 0. Ou seja, a afirmação é incorreta.
9 print('3 é par?', eh_par(3))
10 # Retorna False para 5, mas NOT o inverte para True, confirmando que ele é um número ímpar.
11 print('5 é ímpar?', not eh_par(5))

```

Preencha o teste de mesa para o código acima.

Memória		Tela
[eh_par]	[eh_par]	(7) 2 é par? True
numero	numero	(9) 3 é par? False
(7) 2	(11) 5	(11) 5 é ímpar? True
(9) 11		

Descreva, resumidamente, o que faz o código: Analisa os valores e determina se eles são pares ou ímpares.

03. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```
1 # Define uma função para a área de um quadrado.  
2 def area_quadrado(lado):  
3     # Realiza o cálculo da área, que será expresso quando a função for chamada.  
4     return lado * lado  
5  
6 # Define uma função para o perímetro de um quadrado.  
7 def perimetro_quadrado(lado):  
8     # Realiza o cálculo do perímetro, que será expresso quando a função for chamada.  
9     return lado * 4  
10  
11 # Determina o valor do lado desse quadrado.  
12 valor_lado = float(input('Lado do quadrado: '))  
13 # Imprime o resultado do cálculo, ativando a função da área.  
14 print('Área do quadrado:', area_quadrado(valor_lado))  
15 # Imprime o resultado do cálculo, chamando a função do perímetro.  
16 print('Perímetro do quadrado: ', perimetro_quadrado(valor_lado))
```

Preencha o teste de mesa para o código acima.

Memória		Tela
[area_quadrado]	[perimetro_quadrado]	valor_lado
lado	lado	(12) 6
(12) 6	(12) 6	(14) Área do quadrado: 36 (16) Perímetro do quadrado: 24

Descreva, resumidamente, o que faz o código: Realiza o cálculo do perímetro e da área de um quadrado.

04. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```
1 # Define a função percentual, que comporta as variáveis valor e porcentagem.
2 def percentual(valor, porcentagem):
3     # Realiza o cálculo que resultará no valor de desconto ou acréscimo a ser aplicado ao preço.
4     return valor * (porcentagem / 100)
5
6 # Decide o valor do preço.
7 pr = float(input("Preço: "))
8 # Arruma-se um valor para a porcentagem.
9 vr_p = float(input("Percentual: "))
10 # Traz a função percentual e usa seu resultado para somar com preço, gerando um aumento.
11 pr_acres = pr + percentual(pr, vr_p)
12 # Retorna a função percentual e usa seu resultado para subtrair o preço, gerando um desconto.
13 pr_desc = pr - percentual(pr, vr_p)
14 # Mostra o preço original, o valor da porcentagem de aumento e o preço final com acréscimo.
15 print(f'R${pr} com acréscimo de {vr_p}% fica por R${pr_acres}')
16 # Mostra o preço original, o valor da porcentagem de desconto e o preço final com desconto.
17 print(f'R${pr} com desconto de {vr_p}% fica por R${pr_desc}')
```

Preencha o teste de mesa para o código acima.

Memória			Tela
percentual	pr	vr_p	(7) Preço: 1200
valor, porcentagem	(7) 1200	(9) 15	(9) Percentual: 15
(7) 1200, (9) 15			(15) R\$1200 com acréscimo de 15% fica por R\$1380 (17) R\$1200 com desconto de 15% fica por R\$1020

Descreva, resumidamente, o que faz o código: Pede um preço e um valor percentual e calcula o desconto e o aumento sofrido em cima desse preço.

05. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```
1 # Define a função que comporta um parâmetro para quantidade de minutos.
2 def minutos_para_horas(qtd_minutos):
3     # Pega o total de minutos e realiza uma divisão inteira para converter esses minutos em horas.
4     horas = qtd_minutos // 60
5     # Descobre o resto advindo da divisão entre o total de minutos e 60.
6     minutos = qtd_minutos % 60
7     # Une as informações anteriores e retorna o resultado da transformação de minutos para horas.
8     return f'{horas}h{minutos}min'
```

```

9
10 # Atribui um valor inteiro para representar a quantidade de minutos.
11 minutos = int(input('Quantidade de minutos: '))
12 # Cria uma variável que recebe o resultado da função e usa o valor da variável "minutos" como argumento.
13 horas = minutos_para_horas(minutos)
14 # Imprime o número de minutos e as horas equivalentes a esse total.
15 print(f'{minutos} minutos são equivalentes a {horas}')

```

Preencha o teste de mesa para o código acima.

Memória			Tela
[minutos_para_horas]	[minutos_para_horas]	[minutos_para_horas]	(11) Quantidade de minutos: 560
qtd_minutos	horas	minutos	(15) 560 minutos são equivalentes a
(11) 560	(4) 9	(6) 20	9h20min
#horas ---->	(13) 9	(13) 20	

Descreva, resumidamente, o que faz o código: Cria uma função que realiza a conversão de uma quantidade de minutos em horas e minutos e imprime isso na tela, mostrando o resultado.

06. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```

1 # Define uma função.
2 def trocar(x1, x2):
3     # Inverte a posição dos parâmetros da função.
4     return x2, x1
5 # Atribui um valor inteiro para o primeiro número.
6 n1 = int(input('Primeiro número: '))
7 # Atribui um valor inteiro para o segundo número.
8 n2 = int(input('Segundo número: '))
9 # Cria uma variável que recebe o resultado a função e tem como argumento o primeiro e segundo número.
10 n1, n2 = trocar(n1, n2)
11 # Imprime o resultado mostrando a inversão de valores entrei os dois números. 1º = 2º e 2º = 1º.
12 print(f'Primeiro {n1}; Segundo {n2}.')

```

Preencha o teste de mesa para o código acima.

Memória			Tela
[trocar]	n1	n2	(6) Primeiro número: 6
x1, x2	(6) 6	(8) 8	(8) Segundo número: 8
(6) 6, (8) 8	# n1, n2 ---->	(10) 8, (10) 6	(2) Primeiro 8; Segundo 6.

Descreva, resumidamente, o que faz o código: Pede dois valores inteiros e inverte a posição desses números, onde o primeiro se torna o segundo número e o segundo, o primeiro.

07. Para o código abaixo, escreva uma linha de comentário fazendo a leitura do comando logo abaixo:

```
1 # Define a função "inveter" que recebe o parâmetro "numero".
2 def inveter(numero):
3     # Pega o último digito do número (unidade).
4     u = numero % 10
5     # Remove o último dígito.
6     numero = numero // 10
7     # Pega o dígito correspondente as dezenas.
8     d = numero % 10
9     # Elimina o dígito das dezenas.
10    numero = numero // 10
11    # Captura o dígito das centenas.
12    c = numero % 10
13    # Desaparece com o dígito das centenas.
14    numero = numero // 10
15    # Coleta o dígito da unidade de milhar.
16    m = numero % 10
17    # Transforma unidade em milhar; dezena em centena; centena em dezena e milhar em unidade.
18    numero_invertido = (u * 1000) + (d * 100) + (c * 10) + m
19    # Retorna o número ao contrário, ou seja, começando do final e indo pro começo
20    return numero_invertido
21
22 # Atribui um valor inteiro que deverá conter até 4 algarismos.
23 n = int(input("Digite um número entre 1000 e 9999: "))
24 # Imprime o número original e esse mesmo número ao contrário.
25 print(f'O inverso de {n} é {inveter(n)}')
```

Preencha o teste de mesa para o código acima.

Memória			Tela
[inveter]	[inveter]	[inveter]	(23) Digite um número entre 1000
numero ---> (23) 7539	u -> 9 / numero -> 753	d -> 3 / numero -> 75	e 9999: 7539
[inveter]	[inveter]	[inveter]	(25) O inverso de 7539 é 9357
c -> 5 / numero -> 7	m -> 7	numero_invertido ->	(18) 9357

Descreva, resumidamente, o que faz o código: Pega um número com quatro dígitos e o mostra ao contrário.

Bom Trabalho!