

Questão 1

(6 pontos)

Dado o seguinte programa em Python:

```
a = 1
b = 2

while a <= b:
    a = a * 2
    b = b + 3
    print( a, b )

x = a + b
if x % 2 == 0:
    y = x // 2
else:
    y = x * 2
print("y =", y )

x = 0
for i in range( 1, b+1 ):
    x = x + y
print("x =", x )
```

Escreva abaixo qual será a saída exata fornecida por esse programa:

| Tela do Computador | Rascunho (rastreo das variáveis) |
|--|----------------------------------|
| <pre>2 5 4 8 8 11 16 14 y = 15 x = 210</pre> | |

Questão 2

(11 pontos)

Faça um algoritmo refinado completo OU um programa em Python que:

1. Leia um valor inteiro n pelo teclado. O programa deve repetir essa entrada até que n seja maior que 1 (um);
2. Escreva na tela uma "caixa" vazada delimitada por asteriscos com n asteriscos em cada lado.

Exemplos de funcionamento do programa:

Exemplo 1:

```
Entre com o tamanho da caixa: 1
O tamanho deve ser maior que 1!
Entre com o tamanho da caixa: -2
O tamanho deve ser maior que 1!
Entre com o tamanho da caixa: 2
**
**
```

Exemplo 2:

```
Entre com o tamanho da caixa: 5
*****
*      *
*      *
*      *
*****
```

Se necessário, use o verso desta folha para resolver esse problema.

```
Leia n
enquanto n < 2:
    Escreva "O tamanho deve ser maior que 1!"
    Leia n
fim_enquanto
i = 1
enquanto i <= n:
    Escreva '*'
    i = i + 1
fim_enquanto
Escreva fim_de_linha
i = 1
enquanto i <= n-2:
    Escreva '*'
    j = 1
    enquanto j <= n-2:
        Escreva ' '
        j = j + 1
    fim_enquanto
    Escreva '*'
    Escreva fim_de_linha
    i = i + 1
fim_enquanto
```

└

```
i = 1
enquanto i <= n:
    Escreva '*'
    i = i + 1
fim_enquanto
Escreva fim_de_linha
```

Mesma solução em Python:

```
n = int( input("Entre com o tamanho da caixa: "))
while n < 2:
    print("0 tamanho deve ser maior que 1!")
    n = int( input("Entre com o tamanho da caixa: "))
```

```
i = 1
while i <= n:
    print('*', end='')
    i = i + 1
print('')
```

```
i = 1
while i <= n-1:
    print('*', end='')
    j = 1
    while j <= n-1:
        print(' ', end='')
        j = j + 1
    print('*')
    i = i + 1
```

```
i = 1
while i <= n:
    print('*', end='')
    i = i + 1
print('')
```

⊥

Outra solução:

```
n = int( input("Entre com o tamanho da caixa: "))
while n < 2:
    print("0 tamanho deve ser maior que 1!")
    n = int( input("Entre com o tamanho da caixa: "))

i = 1
while i <= n-1:
    j = 1
    while j <= n-1:
        if j == 1 or i == 1 or j == n or i == n:
            print('*', end='')
        else:
            print(' ', end='')
        j = j + 1
    print('')
    i = i + 1
```

Questão 3

(9 pontos)

Escreva um programa em Python que peça ao usuário para entrar com duplas de valores (Quantidade, Preço Unitário) relacionados com uma determinada compra de vários produtos. Quando o usuário digitar o valor 0 para a Quantidade, o programa deve calcular e exibir na tela o número total de tipos de produtos comprados, a quantidade total de itens comprados e o valor total da compra. Suponha que o usuário nunca irá digitar algum valor negativo (ou seja, não é necessário verificar se o valor está correto).

Exemplo de funcionamento do programa:

```
Entre com a Quantidade:      2
Entre com o Preço Unitário: 10.00
Entre com a Quantidade:      1
Entre com o Preço Unitário: 15.00
Entre com a Quantidade:      3
Entre com o Preço Unitário:  3.50
Entre com a Quantidade:      1
Entre com o Preço Unitário: 50.00
Entre com a Quantidade:      0
Entre com o Preço Unitário:  0

Número total de produtos:  4
Quantidade total de itens: 9
Valor total da compra:     102.50
```

Use o espaço abaixo para resolver esse problema.

TotProdutos = TotItens = TotValor = 0

```
Qtde = float( input("Entre com a Quantidade:      "))
PrecoUnit = float( input("Entre com o Preço Unitário: "))
while Qtde > 0:
    TotProdutos = TotProdutos + 1
    TotItens = TotItens + Qtde
    TotValor = TotValor + Qtde * PrecoUnit
    Qtde = float( input("Entre com a Quantidade:      "))
    PrecoUnit = float( input("Entre com o Preço Unitário: "))

print("Número total de produtos:  ", TotProdutos )
print("Quantidade total de itens: ", TotItens )
print("Valor Total da compra:      ", TotValor )
```

Questão 4

(9 pontos)

Escreva um algoritmo refinado completo ou um programa em Python que efetue a leitura de três valores para os lados de um triângulo, A, B e C. O algoritmo deverá verificar se os lados fornecidos formam realmente um triângulo (ou seja, cada lado deve ser menor que a soma dos outros dois lados). Se os três lados de fato formarem um triângulo, o programa deverá escrever na tela a classificação do tipo de triângulo formado: isósceles (dois lados iguais e um diferente), escaleno (todos os lados diferentes) ou equilátero (todos os lados são iguais).

Exemplos (4 execuções diferentes do programa):

```
Entre com os valores de A, B e C: 1 1 5
Não é um triângulo.
```

```
Entre com os valores de A, B e C: 1 1 1
Triângulo equilátero.
```

```
Entre com os valores de A, B e C: 1 0.5 1
Triângulo isósceles.
```

```
Entre com os valores de A, B e C: 1 2 2.5
Triângulo escaleno.
```

Use o espaço abaixo para resolver esse problema.

```
Leia A, B, C
se A >= B+C ou B >= A+C ou C >= A+B
    Escreva "Não é um triângulo."
senão se A = B e B = C
    Escreva "Triângulo equilátero."
senão se A = B ou A = C ou B = C
    Escreva "Triângulo isósceles."
senão
    Escreva "Triângulo escaleno."
```