Nome: Pedro Luis Teixeira RA: 1460281723036

7. Lista II, Exercício 1, porque é melhor perguntar se o triângulo é equilátero, antes de perguntar se é isósceles.

Acredito que antes de verificar ~~se~~ tipo do triângulo é ~~nesse~~ necessário o programa verificar se as medidas passadas formam um triângulo

```
IF (A>=B+C) OR (B>=A+C) OR (C>=A+B):
    PRINT('Valores digitados não formam um triângulo.')
```

Após essa condição ser verificada é melhor perguntar se é equilátero, pois é a ~~for~~ forma mais fácil de ~~verifi~~ classificar um triângulo é vendo se todos os ~~lados~~ são iguais, se todos forem classifique o mesmo como equilátero, eliminando assim as outras checagens.

```
IF (A==B) AND (B==C):
    PRINT('Triângulo Equilátero.')
```

12. Lista III, exercício 4, explique a resolução feita pelo prof

```
N = INT(INPUT('Digite o valor de N: '))   ← Recebe o valor de quantos números terá na tabela
A, B = 1, 1   ← A e B recebem o valor de 1, que são os primeiros elementos.
K = 1   ← contador é igual a 1
WHILE K <= n-2:   O N-2 é porque os dois primeiros elementos não precisam ser calculados. Enquanto K (contador) for menor, que N-2 o loop repete
    A, B = B, A+B
    ↳ 'A' recebe valor de 'B' e 'B' valor da soma A+B.
    K = K + 1   → é somado ao valor de 'k' mais 1.
```

PRINT (B) → IMPRIME VALOR DE 'B'!

13. LISTA III, EXERCÍCIOS 5, EXPLIQUE A RESOLUÇÃO FEITA PELO PROF

A= INT(INPUT('VALOR DE A: ')) VARIÁVEL 'A' RECEBE UM VALOR INTEIRO, BEM, COMO A VARIÁVEL 'B'
B=INT(INPUT('VALOR DE B: '))

WHILE A % B != 0: → ENQUANTO O RESTO DA DIVISÃO 'A' POR 'B' FOR ZERO O LOOP SERÁ EXECUTADO.

A, B = B, A % B → VARIÁVEL 'A' RECEBE VALOR DE 'B' E 'B' RECEBE O RESTO DA DIVISÃO DE 'A' POR 'B'.

PRINT(f'MDC = {B}') ~~APÓS FEE CONCLUI~~ SE O RESTO DA DIVISÃO DE 'A' POR 'B' FOR ZERO ENTÃO O PROGRAMA SAIRÁ DO LOOP E IMPRIMIRÁ O VALOR DE 'B'!

16. LISTA IV, EXERCÍCIO 3, EXPLIQUE A SOLUÇÃO FEITA PELO PROF.

FROM RANDOM IMPORT RANDINT → É A FUNÇÃO QUE GERA ALEATORIAMENTO NÚMEROS INTEIROS.

V1 = [ ]
V2 = [ ]
V3 = [ ]

TRÊS LISTAS SÃO CRIADAS: 'V1', 'V2' E 'V3'!

FOR K IN RANGE (10): → O LOOP EXECUTARÁ 10 VEZES
X = ~~RANDOM~~ RANDINT (1, 100) → VARIÁVEL 'X' RECEBE UM NÚMERO INTEIRO DE 1 A 100.
V1 . APPEND (X)
V3. APPEND (X) → LISTAS V1 E V3 RECEBEM VALOR DE 'X'.
X = RANDINT (1, 100) → VARIÁVEL 'X' RECEBE UM NÚMERO INTEIRO DE 1 A 100.
V2 . APPEND (X)
V3. APPEND (X) → LISTAS V2 E V3 RECEBEM VALOR DE 'X'.
PRINT ('V1:', V1) IMPRIME AS LISTAS V1, V2 ~~E~~ V3, COM SEUS RESPECTIVOS 10 NÚMEROS. E V3 IMPRIME, COM SEUS ~~RESPE~~ 20 NÚMEROS.
PRINT ('V2:', V2)
PRINT ('V3:', V3)

44. LISTA XIV PYTHON PARA ZUMBIS. PDF. A REPRESENTAÇÃO É IGUAL, MAS O RESULTADO É FALSE. EXPLIQUE PRINT (3='3')

O RESULTADO É FALSE, POIS DE UM LADO TEMOS UM NÚMERO INTEIRO E DE OUTRO TEMOS UMA STRING.

PRINT (3='3')

NÚMERO INTEIRO <— —> STRING.

48. LISTA XIV PYTHON PARA ZUMBIS. PDF. PÁGINA 3, LADO DIREITO, O PROGRAMA É WHILE TRUE, MAS NÃO ENTRA EM LOOPING INFINITO, EXPLIQUE.

```
1 A = 0
2 WHILE A < 3:
3     WHILE TRUE:
4
5         PRINT ('X', END = '')
          BREAK
6     PRINT ('O', END = '')
7     A = A + 1
```

O PROGRAMA NÃO ENTRA EM LOOP, PORQUE NA LINHA 7 EXISTE UM CONTADOR, QUANDO ESSE CONTADOR ATINGE O VALOR 3, O PROGRAMA SE ~~EXEC~~ ENCERRA.

65. LISTA 13. EXERCÍCIO C. EXPLIQUE A SOLUÇÃO DADO PELO PROFESSOR. PARA QUE SERVE A FUNÇÃO LAST

A FUNÇÃO LAST É UMA FUNÇÃO DE ORDENAÇÃO DIFERENTE DA ORDENAÇÃO PADRÃO.

Primeiro é recebido os valores a serem testados e ordenados pelo próprio programa. Depois as tuplas são criadas, conforme a lista dada pelo próprio código, Depois a função LAST ordena essas tuplas ~~de~~ pelo último numero de cada tupla, do menor para o maior, crescente, ~~ao~~ Por fim o resultado da lista final é testado com ~~aque~~ a que é esperado.

67. Lista 14. Exercício E. Explique o código da solução dada pelo professor.

Após recebido a frase a ser modificada pelo próprio código. Depois ~~cada~~ cada palavra é verificada e é ~~formada~~ formada dela uma nova apenas com as letras diferentes, depois de formada as palavras são ordenadas de forma crescente. ~~Por fim~~ Depois de ordenada ~~em~~ no fim de cada palavra é colocado um ' ', ou seja, um espaço para que forme uma frase novamente. Por fim é feito o teste da nova frase, com a frase de parâmetro oferecido pelo próprio código.

70. Lista 14. Exercício H/ explique o código da solução dado pelo professor. Que outras formas, todas mais complicadas e com mais linhas, é possível resolver esse exercício.

Após recebido as palavras a serem checadas do próprio programa a função sorted ordena as duas palavras de forma crescente, se elas se correspondem então elas são anagramas e ~~retu~~ é testado com resposta do próprio código. Se elas não forem também são testadas

Segue outra solução

Pensei em outra duas possibilidades. A primeira parecida com a do prof, porém as palavras são ordenas de forma decrescente. A seguir:

```
def anagrama(s1, s2):
    s1 = [] s2 = []
    p1 = p2 = ''
    s1, s2 = sorted(s1), sorted(s2)
    for c in range(len(s1)):
        p1, p2 = p1 + s1[c], p2 + s2[c]
    s1, s2 = p1[::-1], p2[::-1]
    return s1 == s2
```

isso está fora 'for'

OUTRA FORMA É VERIFICAR O TAMANHO DAS PALAVRAS E VERIFICAR LETRA A LETRA SE SÃO IGUAIS SE UMA DELAS FOR DIFERENTES "L=0" VAI RETORNA FALSE, COMO A SEGUIR:

```
DEF ANAGRA (S1, S2):
  L=0
  IF LEN(S1) == LEN(S2):
    FOR C IN RANGE (LEN(S1)):
      FOR C IN RANGE (LEN(S2)):
        IF S1[C] == S2[C]:
          L+=1
    IF L==0:
      RETURN FALSE
      BREAK
  IF L!=0
    L=0
    RETURN TRUE
  ELSE: RETURN FALSE
```

38. TWP365 REVISÃO ~~FUNÇÕES DE STRINGS~~ FUNÇÕES QUAL O PROBLEMA DE VOCÊ TER CÓDIGO REPETIDO. NAS FUNÇÕES MOSTRADAS NO VÍDEO PODEMOS TROCAR O RETURN POR PRINT, SIM OU NÃO. EXPLIQUE A DIFERENÇA.

ALÉM DO CÓDIGO FICAR LONGO, COMPLEXO, AO TENTAR ENTENDÊ-LO, OU CORRIGI-LO TORNARÁ-SE UMA TAREFA EXTREMAMENTE DIFÍCIL. NÃO PODEMOS DIZER QUE RETURN E PRINT SÃO AS MESMAS COISAS, TÃO POUCO SUBSTITUIR UM PELO OUTRO, POIS O PRINT APENAS EXIBE UM VALOR NA TELA, ENQUANTO O RETURN DEVOLVE UM VALOR PARA A FUNÇÃO QUE O CHAMOU.