

Lista exercícios

Este trabalho é uma breve lista de exercícios para ajudá-los na preparação para a prova, e por isso é um pouco diferente dos trabalhos anteriores.

Vocês devem tentar resolver os exercícios sem o computador, que deve ser usado apenas para verificar se as suas respostas estão corretas.

Exercício.

Sem usar o comando **for**, escreva um programa que apresente os números de Armstrong entre 000 e 999. 153 é um número de Armstrong porque a soma dos seus dígitos elevados ao cubo é igual ao próprio número: $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

Exercício.

Escreva um programa que leia três números do teclado e “printe” os mesmos três números em ordem crescente.

Exercício.

Observe o programa a seguir.

```
import math
import locale

x=0.2
while x <= 2:
    print locale.format('%4.1f',x), \
          locale.format('%6.4f',math.sqrt(x))
    x = x + 0.2
```

Sabendo que sua execução produz

0.2	0.4472
0.4	0.6325
...
2.0	1.4142

apresente uma nova versão do mesmo programa substituindo o comando de repetição **while** por um comando **for**. O resultado final deve ser idêntico. Dica: o range do Python só funciona com números inteiros.

Exercício.

Escreva um programa Python que mostre todos os triângulos retângulos cujos lados a , b , c tenham por comprimento números inteiros entre 1 e 50. O programa deve mostrar os valores para cada combinação encontrada e o número de triângulos. Não é preciso eliminar as soluções simétricas. Formato do resultado:

```
a=3 b=4 c=5 é um triângulo retângulo
...
a=5 b=4 c=3 é um triângulo retângulo
...
a=10 b=6 c=8 é um triângulo retângulo
...
Encontrados xx triângulos
```

Exercício.

É possível obter uma aproximação de Π pela fórmula

$$\sqrt{12 \cdot \left(1 - \frac{1}{4} + \frac{1}{9} - \frac{1}{16} + \frac{1}{25} \dots\right)}$$

Quanto maior o número de termos utilizados no cálculo, melhor será a precisão da aproximação obtida. Escreva uma função que retorne uma aproximação de Π de tal forma que o algoritmo de aproximação use os termos de 1 até i , de tal forma que a diferença (módulo) entre o termo i e o termo $i+1$ não seja maior que um certo valor fornecido como parâmetro.

Exercício.

Escreva um programa que produza uma lista com todas as datas entre 01/01/1900 e 31/12/2015.

Simplificadamente, observe que todos os anos múltiplos de 4 são bissextos e leve em conta os meses de 30 e 31 dias.

Exercício.

Escreva um programa que “printe” na tela os números menores que 20.000 que preservem a seguinte lei de formação:

1 2 4 7 11 16 22 29 ...

Use, obrigatoriamente, algum comando de repetição.

Exercício.

Considere o programa a seguir, exatamente do jeito que está.

```
for a in range(2,21):
    for b in range(2,21):
        print a,"x",b,"=",a*b
```

Escreva uma nova versão desse programa substituindo os comandos “for” por comandos “while”, tal que os resultados produzidos nas duas versões sejam idênticos.

Exercício.

Considere o programa a seguir, exatamente do jeito que está.

```
for a in range(15,1,-1):
    for b in range(2,21):
        print a,"x",b,"=",a*b
```

Escreva uma nova versão desse programa substituindo o primeiro comando "for" por um comando "while", tal que o resultado produzido nas duas versões seja idêntico.

Exercício.

Analise o programa abaixo.

```
a = input("a? ")
b = input("b? ")
while a <> b:
    if a > b:
        a = a - b
    else:
        b = b - a
print a
```

Indique o que é exibido quando:

- a) Os valores digitados são 24 e 13
- b) Os valores digitados são 0 e 0.

Exercício.

Analise o programa abaixo.

```
for k in range(0,10):
    a = 19
    soma = 2
    while a < 10:
        soma = soma + k
        a = a - k
print soma
```

Indique o que é exibido pelo programa

Exercício.

Analise o programa abaixo.

```
a = 1
b = 1
while a < 10:
    print b
    a = a + 1
    b = b * 10
```

Indique o que é exibido pelo programa

Exercício.

Analise o programa abaixo.

```
a=5
while a > 5:
    c = a * a
    for k in range(1,12):
        print k * c
print a
```

Indique o que é exibido pelo programa