

Primeiros passos programação II-python

Continuando as tarefas de introdução à programação, vamos a mais alguns casos.

1. Uma lista está em ordem crescente? Dada uma lista numérica, como por exemplo [1, 7, 23, 55, 61, 61, 70, 99], a pergunta é se ela está em ordem crescente ou não. Olhando a definição diz-se que a lista X está em ordem crescente sse

$$x_i \geq x_j \quad \forall \quad i > j$$

É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

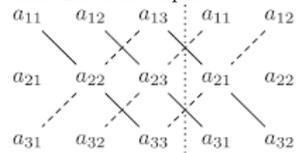
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
    return sentinela
print(orcre([1, 7, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3

Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

```
>>> import numpy as np
>>> a = np.array([[1,2,3],[6,7,8],[10,11,12]])
>>> print(a)
[[ 1  2  3]
 [ 6  7  8]
 [10 11 12]]
```

Este exercício pede que você receba uma matriz assim formada (3 x 3), calcule e devolva o valor do determinante da matriz. Use a regra de Sarrus, como mostrado aqui:



```
def sarrus(x):
    a1=x[0,0]*x[1,1]*x[2,2]
    a1=a1+x[0,2]*x[1,0]*x[2,1]
    a1=a1+x[2,0]*x[0,1]*x[1,2]
    a2=x[0,2]*x[1,1]*x[2,0]
    a2=a2+x[0,1]*x[1,0]*x[2,2]
    a2=a2+x[0,0]*x[2,1]*x[1,2]
    return a1-a2
import numpy as np
a=np.array([[2,4,5],[3,7,0],[1,1,5]])
print(sarrus(a))
print(np.linalg.det(a))
```

Note que o valor pode ser verificado com o comando `linalg.det(x)` que já está programado em numpy e vale para qualquer matriz quadrada.

3. Dia mais frio Suponha que você tem uma coleção de 10 temperaturas mínimas em Curitiba referentes aos primeiros dez dias do mês de julho último. A pergunta que se quer responder é qual foi o dia mais frio nessa dezena. Então, primeiro precisa-se descobrir qual a temperatura mínima da dezena e depois em que dia isso ocorreu. Veja

```
def dmfrio(x):
    i=0
    mini=99999999
    while i<len(x):
```

```
    if x[i]<mini:
        mini=x[i]
        dmin=i+1
    i=i+1
    return dmin
print(dmfrio([12,17,13,14,8,19,26]))
```

4. Quantos estados ? No modelo tributário brasileiro, os estados se endividam junto à União, e mesmo quando o agente emprestado é outro, a União fica como avalista. Desse modo, vários devem muito dinheiro à União. Suponha uma lista com n valores correspondendo à dívida de cada estado e do DF com o Governo Central. n pode ser menor que 27, já que há estados que nada devem. A pergunta a responder é quantos estados respondem por 50% da dívida ? Para responder, primeiro há que ordenar a lista em ordem decrescente, somar tudo e depois ir contando até ultrapassar a metade desse valor. Observe

```
def metadiv(x):
    x.sort(reverse=True)
    met=sum(x)/2
    som=0
    i=0
    while som<met:
        som=som+x[i]
        i=i+1
    return i
print(metadiv([10,20,5,15,49]))
print(metadiv([1,2,3,4,5,6,7]))
```

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de $100m \times 100m$. Em cada célula dessa um número real indica a chance de encontrar petróleo cavando no centro da célula. A Petrobras achou uma sonda que é capaz de extrair o óleo da célula onde está e dos seus até 8 vizinhos. Supondo que você tem a matriz (imagina 8 linhas por 10 colunas) com tais números, a pergunta é onde (L,C) pôr a sonda ?

```
def sonda(x):
    maxi=-9999999
    i=1
    while i<len(x)-1:
        j=1
        while j<len(x[0])-1:
            soma=x[i-1,j-1]+x[i-1,j]+x[i-1,j+1]
            soma=soma+x[i,j-1]+x[i,j]+x[i,j+1]
            soma=soma+x[i+1,j-1]+x[i+1,j]+x[i+1,j+1]
            if soma>maxi:
                maxi=soma
                mai=i
                maj=j
            j=j+1
        i=i+1
    return [mai+1,maj+1]
import numpy as np
a=np.array([[1,4,3,5],[7,4,3,1],[10,1,1,2],[7,3,2,8],[9,8,2,1]])
print(sonda(a))
```

6. Jambo e Ruivão Há dois insetos na natureza: O jambo que tem um ciclo de vida de 13 anos (passa 12 anos hibernando e surge aos milhões no 13º ano) e o ruivão cujo ciclo é de 17 anos (idem idem). O jambo se alimenta preferencialmente de ruivões e quando não os encontra tem muita dificuldade de sobreviver: mais de 90% dos jambos que nascem não chegam à idade adulta por falta de comida. A pergunta é de quanto em quanto tempo, há uma explosão de jambos ? Obviamente isso só vai ocorrer quando ambos nascerem no mesmo ano.

```
def jambo(x,y):
    qt=x
    while 1==1:
        if qt%y==0:
            return qt
        qt=qt+x
    print(jambo(13,17))
    print(jambo(8,6))
```

Para você fazer

Para os exercícios 1 a 6, você deve executar os programas acima sobre a seguinte coleção de dados

- 8 11 25 41 43 48 54 65 82 87
- 28 1 8
9 28 5

- 14 12.9 11.8 15.1 15.5 11 20 10.4 15 12
- 479 706 54 891 619 592 126
- 318 627 384 250 597 189 149 787
717 659 636 591 114 445 195 974
802 234 167 557 397 422 776 229
911 366 699 238 885 130 697 214
82 891 206 474 943 618 838 538

6. 13 10

Para os 3 programas a seguir, você deve construir o programa pedido e depois executá-lo com os dados entregues, achando o número em questão e também entregando a listagem do programa que você escreveu.

7. Maiores que os vizinhos A questão é contar quantos elementos em uma lista numérica dada são maiores que os seus vizinhos. Note que o primeiro e o último elementos não tem um dos vizinhos e fica a pergunta: o que fazer com eles ? Cada problema dá a sua solução. Neste caso, os elementos inicial e final não serão testados, isto é: devem ser desconsiderados. Execute seu programa com estes dados

7. 365 757 704 242 156 347 544 872 961

8. x é primo ? Da definição matemática, o inteiro positivo x é primo se ele é maior do que 1 e só tem 2 divisores: a unidade e o próprio x . Implementando exatamente esta definição, como ficaria ? Execute seu programa com o número

8. 329801

9. Funcionário da quinzena Os 11 funcionários de uma lanchonete do Bloco de Exatas de importante universidade são identificados por um número único e fixo: 1..11. A cada dia a lanchonete escolhe o funcionário do dia (aquele que foi melhor avaliado nesse dia) e registra seu número em uma lista. No final da quinzena, essa lista contém 15 números entre 1 e 11. A pergunta que se quer responder é qual o funcionário da quinzena ? Ou seja, aquele que mais vezes foi citado. Se houver empate, deve ser mostrado o que tiver o maior número.

9. 7 6 10 2 2 6 11 3 3 4 10 4 9 6 1

Responda aqui:

1) cresce?	2) det	3) dia frio	4) quantos est?
5) L. petr	5) C. petr	6) jambo	7) > viz?
8) primo?	9) func?	////// ////// ////// ////// //////	////// ////// ////// ////// //////

observações

- Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificadas com o seu nome e com o número 1
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



===== 04/12/2019 10:59:23.0 =====E=PL049q

1 1 4295 8 2 4 6 130 1 1 6