

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:

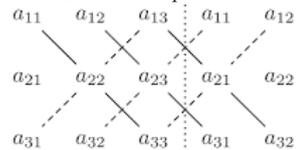
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(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(dmfrío([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 com a 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=-99999999
    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 i==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

- 4 11 20 24 26 27 38 99 73 84
- 7 8 3
9 32 6
5 7 5
- 16 12.8 19.4 17.6 16 14 14.4 11 12 15.4
- 343 239 575 387 470 133
- 364 328 743 800 951 284 811 32 671 575
231 41 851 336 746 171 172 653 112 363
448 504 478 144 838 601 904 386 891 957
197 454 82 374 628 863 353 512 768 674
473 702 64 923 375 654 919 502 622 177

6. 7 8

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. 680 184 255 120 941 309 543 729

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. 208443

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. 8 6 4 9 2 4 11 9 6 9 9 1 1 11 10

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 identificada com o seu nome e com o número 1 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

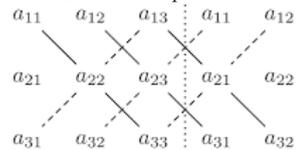
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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

- 7 8 37 49 54 62 71 81 82 87
- 35 3 4
6 15 3
4 2 8
- 13 13.6 13.5 18 14 16.1 19 18 12.5 14.5
- 244 923 940 15 912 298 627
- 777 768 828 367 744 950 705 734 923 22
588 680 653 372 421 88 691 975 366 908
971 301 214 611 739 190 154 820 995 568
429 360 918 460 681 573 999 666 330 741
844 807 656 875 307 637 576 62 885 221

6. 8 5

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. 13 239 797 343 131 390 99 124 210 761

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. 1929331

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. 4 9 6 5 11 7 9 3 2 3 10 9 3 9 9

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 identificada com o seu nome e com o número 2 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

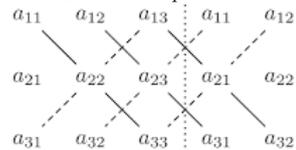
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 4 5 15 23 28 34 51 54 60 90
- 10 5 4
5 25 6
3 1 42
- 13 14.5 17 12 14 17.4 15 13 12.4 18.3
- 424 823 196 757 951 450 657 523
- 268 393 42 587 16 979 59 19
419 966 820 518 184 609 886 128
586 427 688 808 142 742 545 542
401 719 912 52 159 899 978 458
848 134 426 656 996 690 228 575

6. 12 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 escrevesse.

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. 78 302 802 311 775 303 453 850

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. 54679

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. 2 4 11 11 8 1 6 2 1 8 1 6 5 10 6

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 identificada com o seu nome e com o número 3 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

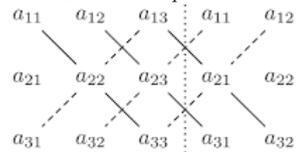
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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:



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 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.

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 com a 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 dmfrrio(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(dmfrrio([12,17,13,14,8,19,26]))
```

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.

```
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 i==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

- 1 4 10 30 40 52 72 73 78 80
- 6 8 9
6 24 6
7 9 7
- 17 16.2 14.1 10.8 14 16 12.3 16 17 15.8
- 330 389 790 843 40 688
- 882 735 68 969 234 721 763 939
540 724 189 571 593 201 350 755
524 333 627 271 324 749 591 992
277 293 368 312 355 619 744 949
371 152 424 562 527 692 398 994

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

- 11 8
- 300 894 292 875 382 920 344 873 215

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

- 1052271

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.

- 1 5 11 3 6 3 2 10 4 3 1 3 1 8 9

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 4 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

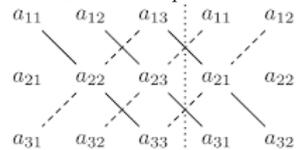
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 6 8 12 18 22 91 79 83 91 94
- 36 5 7
9 42 7
3 9 12
- 18 12 12 14 18.8 18.4 16.2 17.6 15 16
- 592 496 171 590 242 76 794 566
- 940 359 609 711 687 47 327 564
726 412 502 939 336 249 565 907
463 649 972 951 174 285 132 712
343 162 840 152 559 10 995 866
985 474 837 389 405 119 185 492

6. 13 11

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. 654 207 417 144 375 731 39 260 729 853

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. 202049

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. 6 10 8 11 9 8 8 4 6 7 7 7 6 10 10

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 identificada com o seu nome e com o número 5 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

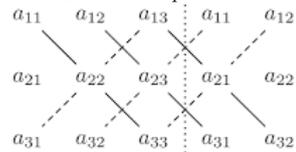
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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:



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 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.

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 dmfrrio(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrrio([12,17,13,14,8,19,26]))
```

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 escrevesse.

```
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 i==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

- 4 5 16 44 51 59 60 95 75 76
- 24 5 3
8 25 5
5 2 24
- 17 12.3 18 19.4 14.6 13 11.8 11 18 13.9
- 862 895 949 636 496 402 972
- 220 278 984 671 949 517 259 632
253 734 533 182 996 50 147 474
692 505 627 336 400 611 640 585
879 398 858 490 756 997 600 356
127 213 708 613 563 619 134 566

6. 15 7

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 escrevesse.

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. 729 932 327 492 50 279 84 725 259 586 470

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. 748487

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. 11 1 7 9 9 10 1 2 1 9 9 4 4 9 5

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 identificada com o seu nome e com o número 6 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

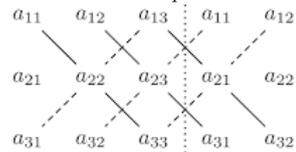
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 1 13 16 29 34 47 78 90 92 99
- 12 1 1
3 32 2
2 5 63
- 10.2 17.2 11.2 17 16 11 18 14 12.5 14.4
- 409 489 948 659 734 945
- 172 521 495 654 390 501 129 712
538 810 120 158 306 870 190 551
191 230 355 43 693 771 905 392
659 356 778 913 791 651 965 903
385 546 161 718 460 362 87 330

6. 12 8

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 escrevesse.

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. 597 235 798 230 984 418 106 406 101

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. 1972349

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. 3 8 9 11 9 8 4 2 9 7 4 11 4 5 4

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 7 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

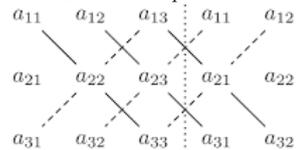
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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(onda(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 i==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

- 1 11 20 24 26 35 39 63 89 83
- 6 2 5
9 30 3
3 2 6
- 15.6 18 16 16.4 12.5 12 15.8 18.4 19 11
- 312 182 438 683 905 448
- 169 99 180 417 889 820 982 327 789
276 102 112 831 701 776 605 130 465
747 241 656 443 615 560 235 414 564
646 845 732 558 556 652 302 385 864
663 509 86 968 974 144 36 464 458

6. 10 6

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 escrevesse.

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. 776 398 888 816 252 340 653 518 599

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. 1889101

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. 3 4 3 9 4 4 9 6 4 1 10 7 5 3 8

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 8 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

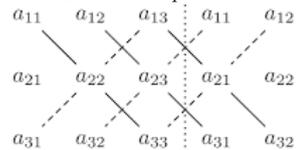
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 com a 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

- 17 18 35 47 90 78 79 90 91 95
- 15 6 6
3 12 6
6 3 16
- 19 20 12.4 12.5 20 18.9 13.8 12 16.9 12
- 228 418 522 565 878 396 500
- 106 660 427 582 156 693 606 131 536
615 474 946 175 125 534 155 480 897
919 643 88 827 886 50 585 192 749
394 485 35 333 592 620 991 632 578
52 511 876 959 17 392 258 400 20

6. 10 5

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. 130 769 998 675 440 351 744 233

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. 1789155

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. 5 2 11 5 8 2 9 7 4 2 11 10 1 11 8

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 9 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

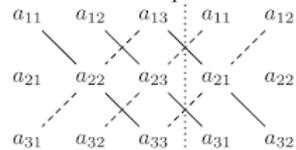
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(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(dmfrío([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 com a 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 i==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 15 20 22 28 33 46 85 99
- 30 8 4
5 24 4
1 7 4
- 12.9 18.3 12 10.7 17 17.3 11 15 10.2 20
- 649 736 129 299 806 323 680 493
- 712 623 998 11 654 341 936 49 640
930 611 207 235 211 173 766 895 356
599 277 937 907 81 285 489 505 515
351 868 138 345 257 468 31 470 876
136 939 267 838 776 975 35 406 258

6. 18 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. 136 630 128 906 289 164 333 179 322 287 62

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. 1301853

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

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 identificada com o seu nome e com o número 10 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

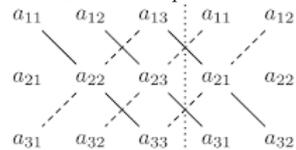
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 10 18 32 33 42 87 78 85 90 99
- 36 9 4
6 14 4
8 8 18
- 20 11 13 13.5 10.2 18 15 11.2 17 14.3
- 770 100 341 332 560 620 851 21
- 297 960 753 743 824 539 720 213 80 512
626 212 306 432 22 50 954 93 808 29
375 366 932 468 999 885 222 510 943 54
671 675 898 107 437 950 166 668 665 813
602 992 87 137 116 284 772 945 40 296

6. 14 8

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 escreve.

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. 222 796 848 511 832 149 256 881 176 877

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. 1525763

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 7 4 1 3 2 3 1 3 10 6 8 2 4 4

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 11 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

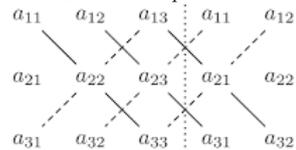
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(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(dmfrío([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 i==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

- 27 28 32 49 87 90 91 94 95 99
- 24 8 5
4 14 2
8 3 16
- 12.9 17 19 17 14.6 13.4 15.5 13.4 16 19
- 585 887 559 495 119 563
- 279 868 884 507 974 883 645 497 203 352
781 223 919 785 383 234 679 839 268 653
772 243 194 84 818 359 564 527 142 896
469 233 251 225 28 420 8 144 82 439
556 281 682 961 292 257 509 154 103 901

6. 18 8

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. 186 892 144 415 264 899 363 894 812

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. 871931

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. 4 5 2 3 1 6 3 3 6 10 9 2 6 7 3

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 identificada com o seu nome e com o número 12 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

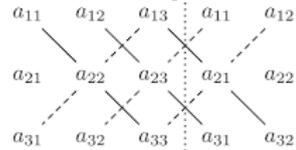
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 com a 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 i==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

- 6 15 94 34 36 41 43 46 50 95
- 42 2 3
4 24 4
4 6 35
- 17.6 19 20 10.1 12 13.3 20 18.2 11 16
- 165 272 711 214 951 402 37 351
- 403 37 229 252 834 56 20 29 704 39
151 568 744 193 95 725 756 621 984 975
718 11 2 873 362 156 772 260 46 564
440 936 942 960 900 385 68 22 402 420
99 413 490 74 919 824 38 73 774 959

6. 12 5

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. 351 449 44 381 947 69 800 228 704 468 764

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. 1268013

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. 5 4 7 5 1 6 10 11 1 4 9 10 3 8 10

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 identificada com o seu nome e com o número 13 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

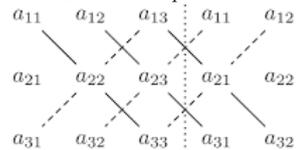
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 9 15 25 28 31 62 85 93 97 98
- 35 2 1
6 35 2
5 5 15
- 10.4 17 11 13.5 17 19 11 12.1 12 14.7
- 847 696 47 505 913 163 671
- 45 586 678 715 633 250 369 366 998
583 963 695 195 770 71 451 221 484
794 351 241 730 29 683 476 384 244
986 634 192 726 85 688 652 51 649
300 814 608 429 563 541 560 92 837

6. 10 11

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. 552 311 621 91 517 725 174 769 989

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. 1378613

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. 5 9 4 8 2 7 10 11 9 5 3 3 6 6 2

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 identificada com o seu nome e com o número 14 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

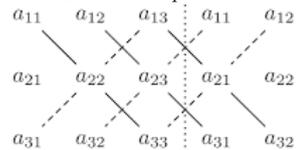
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 com a 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 i==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

1. 2 3 7 10 11 26 42 93 54 70
2. 35 8 4
6 45 7
9 8 30
3. 15 20 11.7 12 19 14 19.5 11.8 16 16.1
4. 640 644 41 653 96 469 142 520
5. 308 8 445 863 464 359 548 478 393 367
839 361 905 683 549 522 328 630 552 790
574 632 105 887 530 896 440 955 399 619
945 736 277 30 129 360 4 903 332 888
796 438 132 210 196 6 310 27 189 917

6. 8 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. 1 972 679 676 841 131 970 372 202 997 43

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. 580871

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. 3 1 7 2 9 2 11 10 6 3 3 4 5 8 4

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 15 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

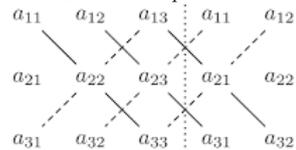
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(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(dmfrío([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 com a dívida. 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 i==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

- 9 88 18 30 52 65 73 75 77 81
- 21 3 7
7 9 6
1 4 35
- 18.7 20 14 15 10.6 20 19 16.7 20 18.8
- 927 58 495 532 366 871
- 31 827 84 42 259 116 51 247 234
373 524 706 453 267 292 809 657 244
877 416 413 497 407 411 861 124 732
501 340 187 953 840 707 440 223 18
192 250 488 65 800 748 803 971 845

6. 15 5

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. 122 360 532 954 175 749 484 18 986 26 905

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. 128485

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. 1 9 7 7 11 6 7 8 8 1 3 6 11 3 4

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 identificada com o seu nome e com o número 16 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

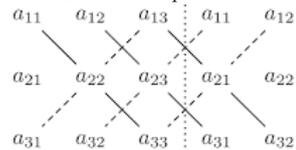
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 com a 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 i==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

- 4 7 26 97 33 34 42 58 72 96
- 24 5 5
5 12 5
7 6 10
- 10.3 14 11.2 17 18.3 14.4 14 17.9 20 15
- 304 930 325 44 740 897 208
- 744 107 426 449 421 547 411 149
747 585 741 214 68 29 480 362
95 233 608 146 535 194 395 230
139 595 853 376 16 176 85 965
529 269 873 413 402 926 953 920

6. 16 9

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 escrevesse.

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. 14 55 458 960 211 19 527 649 919 765 971

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. 540253

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. 1 10 8 7 1 9 1 6 1 10 1 3 6 5 11

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 identificada com o seu nome e com o número 17 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

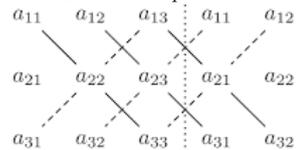
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 5 8 15 29 32 88 63 72 84 90
- 25 5 9
8 28 9
8 6 10
- 19.1 18 20 18 17.8 14 12.1 18.6 15.1 14
- 537 741 350 870 880 277 964
- 92 56 902 434 111 531 653 955 124 674
49 620 341 146 622 584 355 216 369 100
984 121 842 667 505 199 91 48 972 340
577 778 66 744 258 410 6 745 780 669
305 156 110 257 625 573 536 488 167 79

6. 14 9

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. 830 700 69 156 769 578 89 114 344 906 320

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. 620551

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. 1 2 3 2 8 1 4 9 11 5 4 1 2 11 8

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 identificada com o seu nome e com o número 18 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

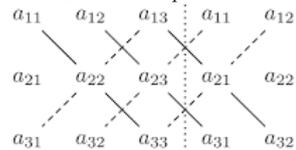
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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(onda(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 i==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

- 2 24 25 31 32 44 61 85 86 97
- 48 6 9
6 48 5
4 6 56
- 17.9 17 17.4 19.4 20 16 19.5 19 11.3 17
- 801 137 248 513 649 484 219
- 387 626 284 297 866 595 721 124 774 72
253 117 381 97 406 233 450 751 977 582
275 16 526 453 348 791 864 49 36 29
875 938 706 921 355 671 404 161 863 768
799 643 104 361 505 573 198 854 360 131

6. 8 5

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 escrevesse.

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. 770 594 937 84 104 743 503 948

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. 623655

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. 8 11 4 5 3 4 3 1 11 3 9 5 6 4 4

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 identificada com o seu nome e com o número 19 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

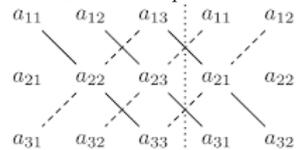
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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(onda(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

- 90 5 6 13 27 70 74 81 89 91
- 12 5 5
9 36 1
8 8 48
- 12.9 16 20 18.7 16 10.4 11 12.4 17.1 19
- 250 1 174 120 531 439 406 986
- 944 787 220 65 876 279 75 728 990 282
754 803 28 827 817 536 962 316 607 373
347 249 421 765 766 33 514 837 978 741
419 714 747 1 43 842 234 454 955 351
584 108 979 152 723 485 457 566 125 160

6. 11 8

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. 339 713 550 322 469 535 919 522 286

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. 642073

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. 3 8 6 7 10 9 1 3 3 9 6 10 2 9 4

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 identificada com o seu nome e com o número 20 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

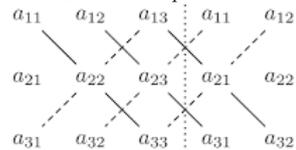
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 12 13 20 21 23 87 34 51 69 78
- 14 5 4
4 49 2
4 7 25
- 13 16.5 15 15.8 13 17 16 18 12.9 18
- 94 852 730 317 343 47
- 344 672 315 235 255 939 563 932 793 426
782 829 377 870 927 345 497 83 40 552
188 752 104 505 768 625 367 589 374 302
655 122 519 171 675 269 531 728 458 590
954 881 749 667 187 238 138 209 686 310

6. 7 8

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 escrevesse.

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. 504 163 322 133 292 927 557 939 175

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. 1495853

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. 5 4 5 10 6 4 2 5 6 10 7 11 8 11 11

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 21 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

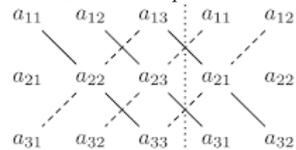
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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(onda(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 i==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

- 10 12 19 46 48 50 62 69 81 89
- 10 2 9
2 3 5
9 1 48
- 20 15 18 19 15.2 14 18 14.2 17.3 19.4
- 517 672 359 302 360 143 810 328
- 62 970 933 341 444 326 926 550 816 712
765 229 989 29 895 852 149 749 106 379
273 192 934 509 55 95 612 984 367 392
981 686 53 179 792 54 909 313 586 888
584 906 954 350 967 955 611 216 795 385

6. 13 6

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. 717 289 506 479 158 711 637 56 840 232

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. 427513

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. 10 8 9 10 8 1 11 5 10 8 7 11 5 8 7

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 22 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

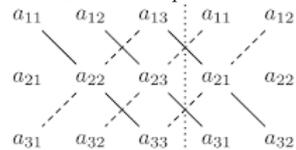
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(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(dmfrío([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 com a 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 i==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

- 42 43 48 50 63 64 69 79 84 86
- 16 7 6
2 28 9
8 9 42
- 14 11 13.2 19 10.1 14 10.9 17 14 15.3
- 948 511 941 274 128 982
- 831 495 170 376 319 856 650 611 917
733 188 449 987 134 955 886 982 316
549 755 326 169 925 728 524 432 594
756 269 189 457 636 740 739 209 912
652 877 657 950 692 512 883 200 869

6. 13 5

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 escrevesse.

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. 165 574 210 348 261 316 405 661

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. 10859

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. 4 11 10 4 1 3 4 8 8 9 5 8 7 5 8

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 identificada com o seu nome e com o número 23 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

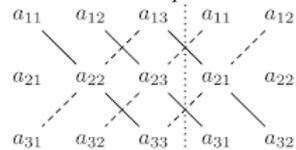
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 89 27 42 52 65 66 70 80 83 86
- 16 8 8
6 18 2
9 7 3
- 13.1 13 14.7 19 18 15 19.9 19.5 14.6 19
- 834 830 272 137 277 498 955 261
- 820 226 96 445 403 624 203 217 565 910
582 518 562 651 294 734 444 289 407 977
42 865 75 829 158 989 459 773 572 391
575 590 873 804 763 883 55 396 288 838
610 484 264 702 691 401 699 166 807 377

6. 7 7

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. 829 142 323 389 520 466 492 605 93

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. 1262519

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. 4 5 1 11 11 4 5 7 10 2 6 1 11 3 9

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 identificada com o seu nome e com o número 24 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

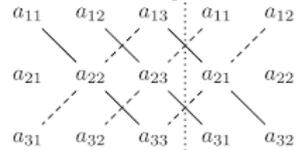
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 6 38 95 63 68 80 92 93 95 97
- 14 7 7
3 12 4
1 2 5
- 18.1 12 17 12.9 15 12 13.8 17.4 10.9 19
- 557 595 817 648 264 725 189
- 268 624 713 97 65 51 728 4 283
384 658 429 150 944 420 194 98 298
118 828 874 540 613 278 889 195 912
449 193 27 820 731 41 790 903 931
961 395 357 537 966 251 149 962 876

6. 13 7

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. 440 36 323 181 976 644 160 349

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. 1495973

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. 3 2 5 1 3 2 10 8 3 8 2 4 9 4 2

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 identificada com o seu nome e com o número 25 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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

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

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:

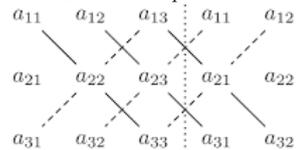
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=+9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 2 11 33 36 43 58 64 79 90 97
- 12 5 2
2 40 3
9 4 15
- 11 13 11.7 14 16.2 14.8 18 18.6 13 19.1
- 726 157 608 425 742 134 406 537
- 526 429 730 147 341 367 701 19 253 802
641 34 619 170 838 24 406 637 801 273
895 127 249 149 334 983 925 12 284 642
617 277 742 640 599 874 84 778 974 30
825 301 537 473 120 294 536 653 456 724

6. 6 6

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 escrevesse.

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. 698 670 600 943 207 555 349 747 308 74 57

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. 1976995

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. 9 3 11 1 1 6 2 11 6 2 4 10 11 11 5

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 identificada com o seu nome e com o número 26 apos 11/09, 50%
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



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:

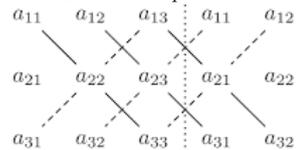
```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
            i=i+1
        return sentinela
print(orcre([1, 7, 2, 55, 61, 61, 70, 99]))
```

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 dmfrío(x):
    i=0
    mini=9999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrío([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 i==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

- 14 31 36 38 42 56 78 79 91 99
- 15 5 1
1 6 6
2 4 48
- 17.7 17.9 19 11.7 19 16 16 11.2 14.9 19
- 32 183 895 49 211 175 753 786
- 195 781 662 648 501 420 850 291
525 304 564 660 450 125 344 390
716 752 323 598 550 175 759 727
907 554 375 502 238 859 892 795
247 693 694 93 60 11 271 949

6. 8 11

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 escrevesse.

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. 513 351 691 152 445 190 148 619 787

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. 609309

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. 6 1 9 4 10 11 3 5 9 11 3 7 5 11 6

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

1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 27 apos 11/09, 50%
2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

