

Primeiros passos programação II-python

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

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

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

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

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

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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 9 27 35 56 59 62 76 80 88
- 2. 24 2 7 4 18 2

- 5 8 12
- 3. 17.9 12 19.9 15 17.5 14.6 18 17.1 15 14
- 4. 401 406 712 802 657 146 595 579
- 5. 232 833 21 847 239 580 35 989 737 753 679 561 391 579 250 725 77 358 820 187 357 507 784 944 95 181 376 661 162 47 769 90 348 924 968 169 640 792 299 171 177 142 514 117 336
- 6. 7 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. 913 84 855 724 137 595 621 771 303 338

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. 569201

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 11 4 1 6 5 10 6 11 10 2 1 1 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

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 1
- 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

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

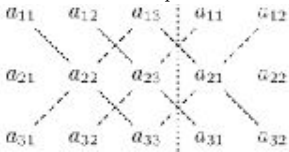
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 3 10 15 17 22 94 52 53 80 94
- 2. 49 4 6
4 20 4
4 4 15
- 3. 19.1 16 13.5 18 11 11.2 18 14 18.7 17.2
- 4. 149 214 942 170 770 981 295
- 5. 274 763 960 544 179 334 912 772 295 418
165 475 616 375 366 498 947 404 204 166
309 98 741 884 329 331 881 592 866 607
953 446 184 228 12 339 828 347 483 629
807 486 941 301 970 141 632 677 710 117

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. 294 933 147 364 393 996 668 505 808 9 55

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. 311821

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

xi ≥ xj ∀ i > j

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

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

2. determinante de matriz 3 × 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 × 3), calcule e devolva o valor do determinante da matriz. Use a regra de Sarrus, como mostrado aqui:



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

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

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

```
def dmfrio(x):
    i=0
    mini=+9999999
    while i<len(x):
```

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m × 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 (imagine 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

- 1. 5 9 13 27 34 38 55 82 85 87
- 2. 20 3 4 1 42 1

- 3. 14.4 11 11.2 14 11 11.1 16.3 18.8 19 17
- 4. 554 738 379 247 899 890
- 5. 520 502 747 412 518 880 862 139 251 836 492 763 915 372 610 44 453 615 544 71 683 868 378 804 182 624 324 260 812 74 743 214 668 997 958 802 216 981 136 655

- 6. 7 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. 157 928 570 278 551 382 134 568 892 446 20

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. 609307

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

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 3
- 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

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

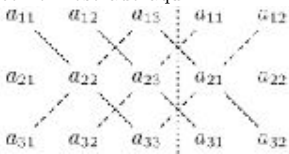
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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

- 1. 2 26 99 34 37 49 70 71 93 97
- 2. 4 3 8
1 12 1
9 5 18
- 3. 11 17 12 17.1 13 17.6 12 15.5 13.1 13.7
- 4. 512 565 298 464 3 645 154
- 5. 611 80 141 246 294 703 877 615 346 49
479 154 522 629 646 695 978 53 552 263
459 768 715 538 25 815 276 831 812 998
219 890 804 50 763 775 389 69 523 312
673 864 658 303 661 885 375 751 509 201

6. 14 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. 163 119 940 342 207 675 412 748

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. 1301851

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 1 8 6 4 2 1 5 1 2 6 7 11 4 1

Responda aqui:

1)crese?	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
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 5 37 60 61 72 76 86 98 92 94
- 2. 40 2 2
6 24 6
3 3 15
- 3. 18 17 12 11.8 14.9 10.8 18 18 14.2 16.7
- 4. 810 584 186 946 403 857
- 5. 286 91 951 396 787 314 697 316 984
800 421 624 370 875 36 46 901 410
849 831 48 737 369 326 837 511 309
939 543 359 135 275 909 604 22 340
371 872 491 670 674 593 997 522 476

6. 14 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. 381 530 521 785 290 631 448 257 176 684 978

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. 1374483

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 3 2 6 10 4 4 2 9 7 1 5 2 10 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 5
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def dmfrio(x):
    i=0
    mini=+9999999
    while i<len(x):
```

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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

- 1. 8 22 27 37 44 47 71 73 74 88
- 2. 15 1 2 5 10 8

- 3. 12.5 14.7 15 18 17 14.6 11 17.3 14 15.5
- 4. 107 523 210 570 589 184
- 5. 200 381 524 147 517 970 718 458 947 573 398 844 446 332 920 491 857 275 695 143 344 721 912 946 134 988 872 190 861 863 351 902 849 472 726 327 618 722 139 257

- 6. 8 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. 110 332 450 87 394 76 839 835

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. 444589

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 10 11 1 5 2 8 1 9 2 11 11 3 9 1

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 6
- 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

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

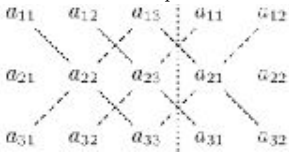
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 28 31 33 34 63 71 80 95 90 96
- 2. 30 7 4
4 42 2
7 6 18
- 3. 13.9 20 15 11.4 11 17 16.8 10.1 16.4 18
- 4. 496 530 596 459 613 417
- 5. 165 724 103 163 898 830 25 664
10 974 295 205 194 514 506 162
533 423 519 879 659 258 319 510
945 146 905 284 203 885 22 32
708 136 957 656 845 112 70 901

6. 17 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. 443 205 213 615 591 890 461 294

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. 520449

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 9 7 7 8 6 1 8 9 2 3 2 9 5 5

Responda aqui:

1)crese?	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
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def dmfrio(x):
    i=0
    mini=+9999999
    while i<len(x):
```

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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

- 1. 1 2 5 20 57 76 83 95 97 97
- 2. 36 7 5
1 12 5

- 4 2 9
- 3. 10.3 12.9 14 15.7 16 20 19 15 16.6 16
- 4. 634 786 179 766 696 383
- 5. 15 400 888 797 290 854 940 104 683
819 717 361 156 302 265 723 411 13
649 323 443 585 621 450 461 884 85
564 512 62 825 455 434 294 499 688
267 177 858 463 238 353 452 667 174
- 6. 18 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. 941 199 374 893 366 108 309 266

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. 763649

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

xi ≥ xj ∀ i > j

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

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

2. determinante de matriz 3 × 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 × 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 × 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

- 1. 1 7 24 90 62 71 78 82 94 99
- 2. 20 9 3 6 24 3

- 9 6 40
- 3. 17.1 14.9 13.4 13 19 11 10.7 15 11 10.1
- 4. 39 640 249 131 374 875
- 5. 977 269 227 71 458 106 763 165 513 806 77 43 97 166 494 211 56 473 220 254 587 741 121 193 187 991 355 623 850 908 338 504 101 662 548 522 429 27 986 557 682 241 794 405 671 385 272 940 559 635

6. 9 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. 145 675 456 18 610 137 355 918 802 983 202

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. 160969

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 8 1 5 4 11 2 4 10 1 8 8 5 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

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 9
- 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

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

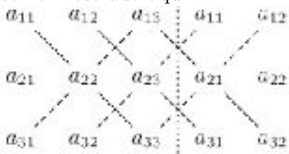
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 2 4 7 36 48 54 62 68 84 85
- 2. 20 6 5
1 63 2
3 1 20
- 3. 15.2 18.9 15 19 19.1 11 17 12.9 18 11
- 4. 784 869 761 505 8 122
- 5. 171 215 62 234 921 928 136 961 664 423
185 826 874 96 939 752 875 613 157 79
896 734 913 31 468 660 957 290 253 786
216 227 489 370 384 306 366 47 295 401
309 12 152 82 292 540 524 150 658 676

6. 6 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. 135 5 884 174 284 634 674 340 679 941 231

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. 1236859

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

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 10
- 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

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

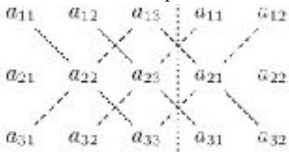
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 21 24 35 36 39 43 64 69 77 99
- 2. 18 8 1
4 6 5
6 4 20
- 3. 15 17 16 12.9 11.4 13.4 11 13 11 18.4
- 4. 881 862 150 788 855 800 18
- 5. 779 372 897 515 797 71 908 484 954 614
281 522 445 345 805 579 387 129 768 924
674 905 642 573 161 812 102 794 675 275
679 164 936 215 183 806 529 578 950 641
73 930 646 212 200 40 746 178 84 416

6. 16 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. 703 283 833 246 477 607 839 313 240

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. 191719

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

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 11
- 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

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

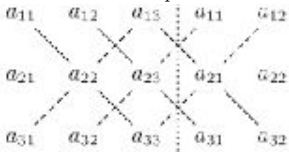
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 10 21 35 39 45 49 93 57 66 68
- 2. 24 8 7
9 6 4
6 8 12
- 3. 14.8 19 15.3 18 12 15.7 20 14.9 20 12.8
- 4. 61 640 981 76 435 243
- 5. 880 664 903 243 221 311 793 548 321
977 431 463 586 199 654 868 22 265
67 555 704 468 917 758 833 438 110
551 246 607 901 145 270 653 33 404
687 981 736 100 689 827 188 478 240

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 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. 455 803 799 646 756 365 263 906

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. 334069

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

Responda aqui:

1)crese?	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 12
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 2 16 18 23 30 35 48 51 90 88
- 2. 36 9 2
3 36 2
5 6 42
- 3. 14 14 19 15.9 12.9 14.5 17 18 15.5 20
- 4. 102 504 319 739 32 571 537
- 5. 115 353 110 988 906 944 816 695
1 981 192 430 961 668 391 713
31 650 338 764 482 465 388 340
325 893 414 795 654 902 184 922
914 175 381 678 3 593 209 171

6. 16 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. 386 647 554 480 262 705 286 148

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. 1640743

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 6 11 5 3 2 8 10 4 6 3 5 11 1 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 13
- 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

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

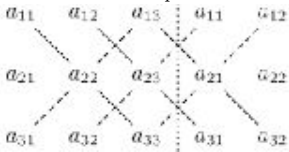
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 3 9 16 28 36 41 51 56 74 93
- 2. 35 7 2
7 21 7
3 4 4
- 3. 13 19.4 17.2 16.3 11 13.3 12.7 17 12 18
- 4. 218 879 348 32 627 179
- 5. 169 562 422 889 651 65 596 210 480
302 327 751 678 259 389 805 463 567
367 552 12 100 379 49 335 212 348
542 476 675 537 178 85 203 371 256
298 637 591 731 884 426 490 162 705

6. 15 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. 539 219 465 146 138 353 522 313 317 618

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. 302143

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 2 8 7 2 4 9 3 2 8 1 6 11 8 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

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 14
- 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

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

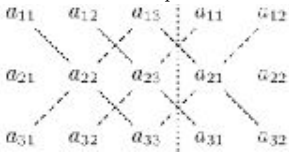
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 10 26 41 51 52 53 58 70 77 96
- 2. 7 1 4
4 6 5
9 5 15
- 3. 16 18.2 13 12.9 14.3 12 18 17 11.2 13.4
- 4. 529 507 956 432 817 28 231 601
- 5. 649 524 226 866 276 878 470 812 63
654 507 405 602 502 320 396 981 74
418 296 378 551 767 612 887 4 321
949 10 268 422 508 69 324 236 169
50 670 279 298 904 722 44 653 273

6. 11 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. 154 588 289 844 663 435 660 958 766 741 814

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. 70139

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 5 8 10 2 7 10 1 4 6 4 1 7 2

Responda aqui:

1)crese?	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
- 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

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

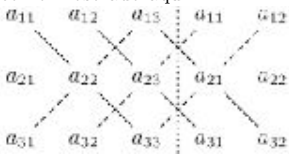
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 18 50 51 61 68 70 75 81 82 86
- 2. 32 3 3
4 8 7
3 3 7
- 3. 15 12.6 11 16.3 14 14 12 13 17 19.1
- 4. 306 485 835 907 344 271 588
- 5. 87 925 238 588 410 194 776 896 505
350 830 466 808 138 555 180 513 817
557 829 542 340 241 759 617 49 649
17 603 257 786 802 202 714 454 880
855 114 558 859 38 66 769 816 370

6. 7 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. 18 245 729 54 483 555 936 775 928

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. 9547

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

Responda aqui:

1)crese?	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 16
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

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

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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 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

- 1. 7 15 22 45 87 78 87 89 94 95
- 2. 12 5 4 4 56 3

- 9 9 36
- 3. 14 16.5 17 13.9 17 16 17.6 13.8 16.2 11
- 4. 250 343 686 445 745 509 958
- 5. 261 604 146 160 383 237 835 742 964 225 445 489 270 235 306 804 887 890 441 31 936 240 527 372 977 1 10 932 754 854 196 526 153 11 912 664 338 626 791 120 846 155 743 587 447

- 6. 17 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. 296 492 531 914 310 175 363 572 293

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. 2243

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 4 11 10 11 9 1 9 2 11 1 3 1 5 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 17
- 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

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

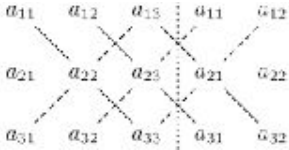
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 4 14 98 39 42 43 45 51 75 89
- 2. 15 7 9
2 6 1
5 4 20
- 3. 18 13 12 16 14.6 10.4 14.5 14.2 14 10.5
- 4. 554 185 129 641 929 361 551
- 5. 48 93 252 510 323 785 142 179 916 698
321 914 921 137 979 972 991 319 841 734
77 541 793 846 892 837 757 684 111 743
177 165 349 935 553 317 462 493 444 526
552 729 603 239 102 789 256 129 917 890

6. 6 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. 789 198 861 726 584 593 871 744 763 916

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. 654427

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 6 6 2 1 9 9 8 7 7 2 6 7 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 18
- 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

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

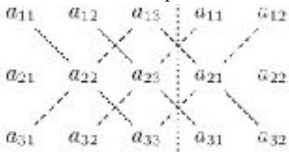
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 6 15 22 30 34 37 44 59 62 93
- 2. 6 4 4
3 63 3
6 3 24
- 3. 18 13.1 18 19.4 11 12.9 17.4 11 15.2 12
- 4. 797 499 901 897 645 675
- 5. 185 764 737 318 481 315 771 65 544
610 835 513 680 105 552 89 197 537
100 371 807 818 394 686 204 359 587
147 310 482 892 497 783 261 360 916
146 331 805 885 72 169 327 449 438

6. 17 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. 21 145 269 811 486 123 857 204

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. 688219

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

Responda aqui:

1)crese?	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 19
- 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

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

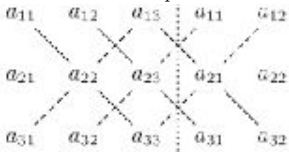
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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 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

- 1. 97 3 15 17 23 49 53 59 90 99
- 2. 36 7 3
2 25 6
5 5 18
- 3. 16.2 14.1 15.2 17 12 12.5 20 16.3 11 16
- 4. 873 88 491 807 545 264 756
- 5. 280 273 979 516 948 327 227 174 807 885
183 85 428 720 900 317 497 932 982 842
207 123 504 873 612 161 588 611 825 39
688 282 240 718 154 685 838 86 615 313
106 14 938 517 284 920 971 627 186 556

6. 9 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. 277 464 818 33 602 627 883 218 924 590 857

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. 77263

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

Responda aqui:

1)crese?	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 20
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def dmfrio(x):
    i=0
    mini=99999999
    while i<len(x):
        if x[i]<mini:
            mini=x[i]
            dmin=i+1
        i=i+1
    return dmin
print(dmfrio([12,17,13,14,8,19,26]))
```

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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 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

- 1. 6 16 20 23 25 47 48 61 75 99
- 2. 15 1 4
1 48 5
6 9 28
- 3. 10.3 10.3 15 18.1 20 14.7 14 16 17.2 11
- 4. 231 283 458 705 623 606 723
- 5. 205 705 663 397 874 891 232 113 988
805 340 779 763 850 771 691 210 213
57 880 548 857 885 654 894 72 984
291 582 395 390 695 794 525 389 182
211 27 902 208 767 715 209 475 240

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 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. 694 151 101 14 928 818 270 253

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. 688217

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

Responda aqui:

1)crese?	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
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

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

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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 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

- 1. 23 35 44 59 64 65 67 73 85 98
- 2. 9 2 4 5 15 3

- 7 3 12
- 3. 11 13 14.2 10.1 19.8 12.5 14 16 13 16.3
- 4. 130 278 854 269 13 220 697
- 5. 494 892 450 479 646 82 888 523 466 413 754 381 14 805 830 115 881 201 730 178 909 384 146 173 354 303 758 704 162 545 935 210 719 428 515 139 176 970 774 129 852 752 927 23 885 594 612 275 528 859

- 6. 17 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. 334 640 83 442 222 635 50 382 359 828 813

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. 1680803

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 4 11 6 4 4 11 2 8 2 6 6 7 3 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 22
- 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

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

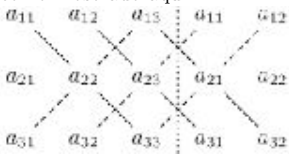
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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 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

- 1. 5 13 36 87 58 63 66 90 95 96
- 2. 21 1 9
7 48 3
7 3 49
- 3. 15 20 15.3 18.5 18.9 14 12.5 14.5 12 18
- 4. 620 142 822 890 223 338
- 5. 455 989 215 17 12 442 270 833 492
101 211 937 264 675 814 20 793 750
495 392 160 71 812 704 103 328 298
651 956 183 502 804 726 161 773 899
333 689 383 691 518 890 479 982 710

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 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. 788 57 720 290 357 286 422 857 5 860 424

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. 8 5 1 2 9 5 1 10 8 3 3 6 1 8 5

Responda aqui:

1)crese?	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 23
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 25 27 29 32 50 51 64 77 87 99
- 2. 15 9 4
5 4 4
2 4 15
- 3. 19.1 10.6 16.9 19 19.3 14 17 14 19.8 11
- 4. 483 757 271 339 648 709
- 5. 610 944 40 22 904 874 78 258 77
582 549 728 203 995 503 741 897 683
686 322 777 493 861 514 762 568 402
532 39 220 789 344 301 498 833 155
869 246 912 84 168 792 611 698 959

- 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. 252 194 914 320 865 210 376 697 235 780

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. 321949

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 5 9 9 2 2 1 2 8 10 1 1 5 6 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 24
- 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

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

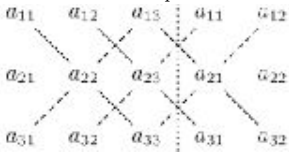
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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 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

- 1. 3 7 8 18 34 82 88 92 95 96
- 2. 3 1 4
6 15 3
3 6 35
- 3. 12.9 16 17 17 19.8 13 13 13 16 19.4
- 4. 598 438 792 65 66 584
- 5. 380 569 743 155 870 835 924 307 75
13 141 104 790 760 788 333 17 948
156 148 35 848 550 184 578 24 658
854 661 971 270 215 495 573 527 116
894 802 822 243 242 673 394 152 596

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. 831 540 604 813 147 819 154 654 802 972 660

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. 971041

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 7 11 11 5 6 8 5 1 3 1 7 3 5 3

Responda aqui:

1)crese?	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 25
- 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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3×3 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 (3×3), 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

- 7 15 27 38 60 62 66 79 90 92
- | | | |
|---|---|---|
| 8 | 8 | 9 |
| 5 | 8 | 4 |

8 7 48

3. 14.1 17.5 18 10.5 11.1 15 17 19 15 16.8

4. 744 981 586 359 63 305 262

5.

496	212	2	206	407	655	390	919	79
366	430	487	30	346	389	708	886	15
103	471	262	340	921	61	952	836	442
908	218	421	770	291	677	573	83	499
885	412	443	786	925	569	567	258	102

6. 15 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. 7 372 100 303 467 459 626 31 898 779 910

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. 1674331

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 6 2 10 10 1 11 4 4 8 9 6 9 1 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 26
- Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



102-75510 - /

Primeiros passos programação II-python

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

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

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

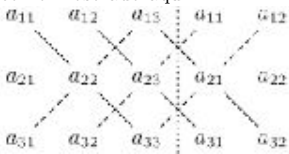
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

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

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

```
def 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 x 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

- 1. 9 10 11 12 15 17 39 54 57 81
- 2. 32 3 1
3 24 2
9 8 28
- 3. 13 20 16 19.4 11.8 12 19.3 12 20 19
- 4. 416 909 68 720 936 595 73 667
- 5. 472 524 103 436 464 826 358 690 948
81 63 350 535 53 678 450 10 108
879 387 418 244 102 662 931 80 576
99 744 474 547 52 540 575 917 329
300 400 356 35 433 226 163 642 945

6. 13 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. 763 624 60 715 349 233 151 519 238

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. 369833

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

Responda aqui:

1)crese?	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
- 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

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

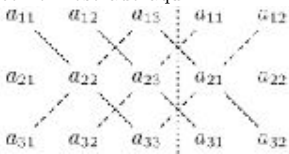
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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, 23, 55, 61, 61, 70, 99]))
```

2. determinante de matriz 3 x 3 Para este exercício precisa-se introduzir o conceito de matriz. É um arranjo bidimensional de dados. Cada elemento vai ser individualizado pelo uso de 2 índices: o da linha e o da coluna, nesta ordem. Uma matriz é criada em Python chamando-se o pacote Numpy (Numerical Python). Acompanhe

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

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



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

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

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

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

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

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

5. Plataforma de petróleo O fundo da baía de Guaratuba foi mapeado em células de 100m x 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

- 1. 7 8 18 21 38 44 73 79 92 92
- 2. 25 4 8
2 15 8
5 9 36
- 3. 11.9 19 13.1 10.7 19 17.3 16 11 19.1 11
- 4. 183 972 611 505 644 311
- 5. 268 925 378 446 430 719 338 916 504
633 552 96 891 309 897 809 603 752
133 599 788 963 769 720 921 724 181
329 721 574 881 821 49 905 345 692
983 944 286 225 780 826 525 990 831

6. 8 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. 991 290 752 544 313 518 120 176 498 558

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. 966781

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 6 1 11 11 8 2 9 5 11 4 2 4 5

Responda aqui:

1)crese?	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 28
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

