

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

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

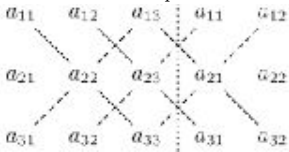
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
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 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. 22 86 46 52 55 71 75 76 88 93
- 2. 24 8 2
2 16 6
5 7 36
- 3. 14.4 17 11 12.3 19 11 12 13 17.2 15.9
- 4. 171 244 541 507 680 364 599
- 5. 309 537 895 157 85 831 653 64 881
944 892 686 18 613 937 94 777 665
555 755 112 676 480 827 523 932 140
273 304 741 864 623 240 926 640 192
89 609 378 249 813 999 306 327 979

6. 10 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. 596 93 256 850 77 469 784 150

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. 1494613

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 1 6 4 5 2 11 11 1 10 4 11 5 6 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 1 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

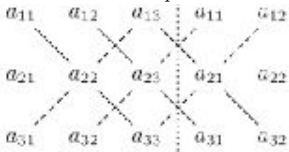
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 86 18 45 58 71 73 76 82 89
- 2. 14 1 9
7 30 1
1 3 35
- 3. 17 19.2 12 10.5 12.3 15.6 14.7 11 18 11
- 4. 132 483 315 643 593 847 200
- 5. 798 69 469 994 715 219 264 415 106 718
458 723 315 700 863 407 299 372 970 704
394 5 660 348 150 810 291 168 476 409
666 8 441 16 156 595 997 200 749 316
642 49 763 501 19 464 228 789 886 558

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. 335 639 629 487 13 616 7 398 83

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. 1366507

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

9. 4 11 8 9 6 5 2 8 6 2 6 8 9 10 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 2 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

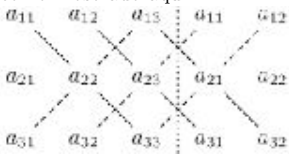
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 6 31 34 38 45 46 52 62 93
- 2. 9 3 3
1 28 6
9 9 25
- 3. 10.4 19 13 12.7 18.3 14.7 12.8 15 13 16
- 4. 439 880 551 508 842 142
- 5. 93 775 698 251 841 321 949 345 138
493 721 587 689 400 329 957 122 743
315 2 325 607 101 637 194 123 232
83 448 120 148 709 813 197 351 97
652 461 909 601 780 20 662 593 654

6. 18 8

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

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

7. 979 530 517 251 77 672 252 402 2

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. 1170835

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 3 7 2 1 3 1 3 11 3 2 3 1 9 10

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 3 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

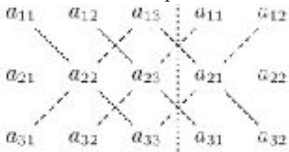
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 7 97 40 49 51 65 71 84 90 95
- 2. 7 3 1
4 21 4
3 1 6
- 3. 14 18.3 17 17 19.1 14 16 19.9 14.2 16
- 4. 434 309 90 622 874 523 649 187
- 5. 893 386 55 684 818 462 200 64
725 716 419 759 480 814 274 667
38 636 905 499 683 665 158 933
909 863 982 88 585 632 357 214
750 348 809 922 135 192 145 900

6. 15 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. 255 575 847 750 76 895 761 160 913

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. 905211

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

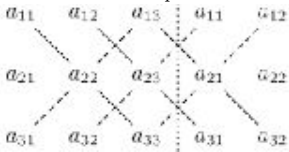
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 9 21 39 42 54 57 93 80 94
- 2. 20 2 1
5 27 3
6 5 30
- 3. 11.3 13.7 15.1 17 12.6 19 16 10.7 17 13
- 4. 882 552 218 698 392 553
- 5. 274 316 86 153 216 27 890 70
235 116 255 217 924 460 442 913
873 935 457 843 121 696 339 938
432 188 688 313 284 712 763 707
64 398 18 266 908 854 958 36

6. 16 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. 202 925 856 547 971 229 269 545 531 736

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. 1941091

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 5 4 5 8 6 2 1 3 1 10 6 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 5 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

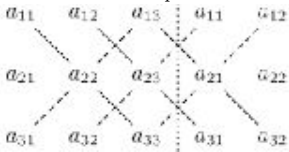
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 15 30 31 39 48 60 65 76 90 98
- 2. 48 7 8
9 20 9
2 6 36
- 3. 18.2 17.9 16 11 19.3 16.1 18.4 11 18 17
- 4. 348 328 898 148 271 902 885 537
- 5. 787 739 902 392 504 852 615 669
319 68 542 385 214 602 29 116
44 464 652 797 964 610 235 384
934 354 226 645 976 168 717 959
448 212 988 38 666 933 803 167

6. 12 8

Para os 3 programas a seguir, você deve construir o programa pedido e depois executá-lo com os dados entregues, achando o número em questão e também entregando a listagem do programa que você 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. 118 726 691 748 896 77 652 102 491

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. 1230067

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

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 6 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

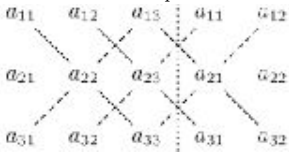
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 1 8 11 13 29 52 56 63 66 78
- 2. 6 6 3
9 6 6
7 2 12
- 3. 14 20 16.2 14.3 13 16 15.6 19 17 17
- 4. 712 175 757 545 308 819 402
- 5. 632 510 215 692 206 613 739 172 39 66
946 296 3 212 453 81 941 835 680 516
55 320 965 255 609 529 110 264 408 509
920 853 675 885 430 754 929 202 94 786
664 111 631 581 157 709 145 932 442 413

6. 18 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. 556 540 120 230 674 510 276 173 288

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. 1121831

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

9. 3 1 10 6 8 9 3 6 6 11 6 4 9 9 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 7 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

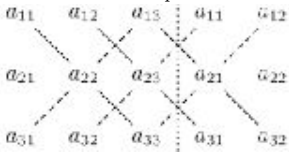
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 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 44 52 58 74 76 78 81 83 97
- 2. 24 6 8
5 63 5
1 6 54
- 3. 19 10.7 10.4 14 20 19 14.4 20 19.8 13
- 4. 626 543 890 454 395 982 968
- 5. 332 768 486 546 625 636 432 98 315
428 361 134 723 250 979 833 39 187
660 28 877 930 89 466 992 948 543
661 713 325 360 891 526 871 691 425
560 568 923 147 791 627 725 109 563

6. 12 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. 234 612 486 880 178 831 863 37 684 828

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. 1801311

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 3 2 1 3 8 5 4 1 10 10 7 8 3 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 8 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

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

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 8 14 37 46 57 97 69 70 91 94
- 2. 56 3 8
4 18 9
2 5 40
- 3. 16.2 13 15 16 15.5 18 15 10.4 15.1 18.2
- 4. 461 329 232 631 661 55 282
- 5. 386 910 780 609 24 359 763 132
447 40 201 161 776 956 466 321
176 564 78 135 816 138 795 671
607 188 684 465 211 438 200 680
732 558 830 903 588 861 227 172

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. 665 384 539 305 464 580 143 431

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. 1910769

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

9. 4 5 10 6 3 4 9 6 1 11 2 6 1 4 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 9 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

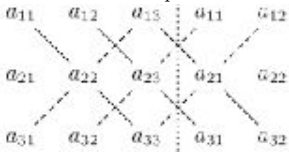
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 12 23 24 27 40 88 44 65 98
- 2. 45 3 1
3 36 6
2 2 18
- 3. 18 15 15.3 15 16 19.2 14 13.7 10.3 15.7
- 4. 708 54 162 835 143 487
- 5. 57 100 978 32 568 4 480 494 704 61
12 2 1 847 200 834 103 356 375 167
443 198 914 193 390 702 333 163 180 752
832 349 818 918 116 924 105 22 258 693
810 447 298 858 971 644 408 920 38 304

6. 8 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. 853 537 217 739 160 785 194 562

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. 574701

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 5 5 4 1 10 4 2 9 10 7 5 1 6 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 10 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

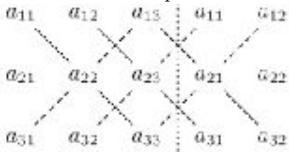
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 1 16 19 23 28 41 63 70 93 97
- 2. 15 7 8
7 15 8
2 2 21
- 3. 14.1 16 14.1 13.2 19 16 15.4 19 14 10.8
- 4. 53 664 925 709 270 872 32
- 5. 836 11 967 540 931 120 524 216
39 202 661 593 453 562 247 72
713 553 408 555 858 477 790 352
550 568 826 409 766 703 336 995
560 37 74 820 502 946 671 28

6. 10 6

Para os 3 programas a seguir, você deve construir o programa pedido e depois executá-lo com os dados entregues, achando o número em questão e também entregando a listagem do programa que você 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. 995 526 476 252 784 852 425 67 551 644

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. 1643351

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 10 9 8 10 1 6 9 10 9 5 10 10 5 4

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 11 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

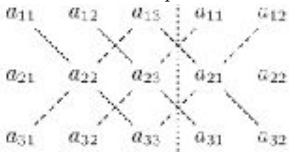
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 96 38 42 54 60 63 66 70 92
- 2. 32 2 9
9 15 7
1 4 35
- 3. 12 12 19 14 12 16 11 14 12.2 14.5
- 4. 538 266 888 456 647 215 109
- 5. 592 546 777 260 186 290 501 653 868 93
303 361 415 453 521 511 893 647 604 481
353 295 435 752 952 400 29 957 76 517
651 960 836 270 73 72 545 578 212 433
626 806 852 239 569 8 222 781 368 140

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. 156 420 573 242 43 970 964 357

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. 1978421

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

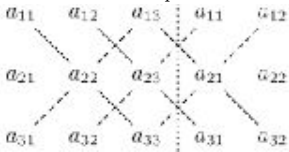
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 8 19 95 55 57 68 77 78 87 92
- 2. 9 8 3
5 18 3
4 7 18
- 3. 18 12.2 11.3 11.3 11.2 19.7 14 19 15 16
- 4. 806 164 380 861 24 759 221
- 5. 771 886 711 692 533 678 238 847 383 765
580 719 305 969 110 467 231 979 514 596
919 353 54 491 990 943 9 30 517 903
859 362 890 506 941 345 727 413 925 74
373 326 109 283 210 576 864 360 154 734

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. 910 656 425 747 391 352 227 645 643 110

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. 943603

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 10 10 5 11 4 7 3 8 8 2 11 2 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

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 13 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

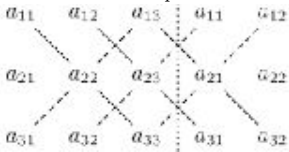
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 1 11 18 20 30 34 35 48 84 97
- 2. 36 6 1
9 12 8
2 6 25
- 3. 11.7 13.5 14.8 16 18 15 13.2 13.8 15 15
- 4. 960 215 170 655 442 20 807 281
- 5. 44 893 123 458 381 658 709 422
807 728 875 168 342 784 467 54
615 272 905 598 76 111 157 683
702 630 291 588 828 378 987 14
483 303 607 664 912 693 36 124

6. 18 10

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

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

7. 135 568 934 187 351 33 288 2 922 726 950

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. 1775573

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 1 10 9 8 8 2 8 6 11 3 1 3 8 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 14 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

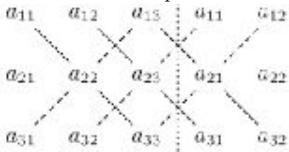
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 quando 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 15 16 28 40 54 57 78 83 97
- 2. 6 6 9
8 54 3
5 1 30
- 3. 17.4 17 15 19.7 17 14.1 17 17.5 17 16
- 4. 655 773 611 301 608 918 455
- 5. 39 179 869 365 108 754 935 530 197
91 462 234 342 773 432 883 943 996
550 255 519 428 545 864 358 562 348
84 164 120 141 186 405 886 335 330
414 741 547 3 86 125 375 237 882

6. 18 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. 611 465 365 712 634 273 282 554 542 113 644

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. 81133

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 1 5 2 2 8 7 11 3 4 6 2 4 9 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 15 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

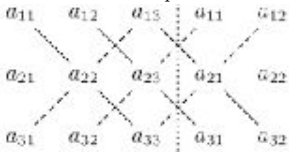
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 18 19 25 28 42 45 47 80 88
- 2. 42 7 7
8 24 8
5 5 24
- 3. 15.8 14.1 16 10.9 18 14 11 13 19.5 19.5
- 4. 60 263 571 622 83 853
- 5. 580 588 540 262 130 694 608 648
38 561 546 916 29 515 210 111
463 615 125 4 181 329 884 620
684 639 771 573 239 732 747 73
270 123 604 183 847 831 387 879

6. 16 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. 831 367 48 422 650 863 87 743

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. 384915

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

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

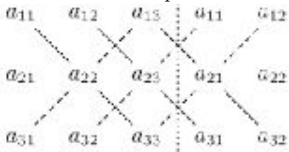
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 quando 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 13 96 37 52 53 63 67 73 97
- 2. 10 5 2
6 14 8
9 3 48
- 3. 12.5 12 17.1 11.1 15.7 15 15.9 16 14 13
- 4. 575 165 881 687 414 492 895 831
- 5. 99 453 468 195 986 83 975 22
699 121 695 257 363 971 371 615
150 262 124 193 6 744 766 472
964 404 917 450 481 638 442 681
645 400 801 110 751 115 747 630

6. 10 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. 113 731 537 934 573 705 527 676

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. 898069

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

9. 8 6 4 2 6 2 11 5 5 4 6 4 5 10 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 17 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

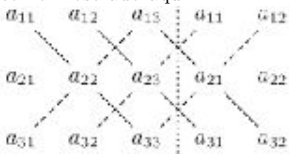
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 7 8 18 29 60 67 70 72 89 99
- 2. 40 7 1
1 54 9
2 3 27
- 3. 20 17 19.5 20 18 19.4 12.6 16.4 11 20
- 4. 595 888 407 477 725 81 503 832
- 5. 834 323 412 5 659 286 792 191 620 189
163 66 259 122 957 213 691 460 605 432
997 65 149 316 872 795 186 905 865 726
768 456 616 325 470 497 696 779 103 42
570 675 239 899 536 217 642 704 855 132

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. 235 368 491 338 320 418 448 681 139

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. 1414425

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 11 1 11 9 7 7 11 5 4 1 9 2 7 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 18 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

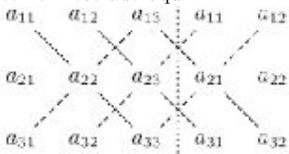
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 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. 22 25 35 42 65 76 79 87 88 89
- 2. 6 4 5
2 54 7
2 8 4
- 3. 15 16.6 13.4 18 15 17 10.6 13.3 11.9 12
- 4. 874 721 636 967 963 538 287 36
- 5. 964 778 410 510 286 82 328 436 184
117 636 134 63 845 703 641 221 296
560 89 652 120 665 821 408 498 387
259 50 437 182 141 147 201 138 683
746 883 298 81 404 601 482 651 102

6. 12 8

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

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

7. 122 610 292 675 281 918 661 813 994

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. 1489223

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 9 6 7 7 2 11 8 9 7 5 2 3 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 19 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

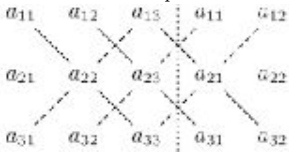
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 86 36 42 43 51 54 79 91 99
- 2. 25 4 3
3 36 7
3 6 28
- 3. 15 13.7 15.7 18 10.2 16 10.6 20 13 13
- 4. 928 223 196 421 164 578 344 500
- 5. 221 52 226 189 612 268 880 948
601 686 148 40 438 350 814 661
407 872 622 129 434 54 435 440
188 689 63 768 594 979 291 836
514 383 937 124 555 842 735 76

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. 605 268 254 507 817 322 671 291 858 725

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. 235513

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

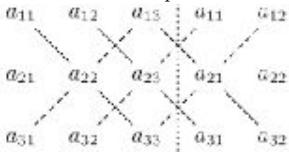
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 7 18 19 43 53 76 99 83 95 99
- 2. 54 4 9
2 30 2
6 4 7
- 3. 17 14 19.7 11.2 17.6 12.9 10.8 13 13 13
- 4. 190 805 258 836 702 778 108
- 5. 587 341 404 259 354 326 875 382
730 428 601 285 291 186 881 50
993 121 719 334 26 958 806 202
721 663 783 660 568 563 529 961
811 553 281 144 255 789 747 133

6. 9 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. 89 897 473 591 775 391 930 942 237 43 756

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. 976601

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

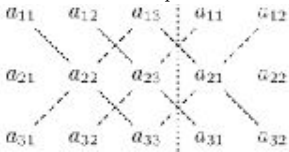
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 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 25 26 48 59 61 69 78 89 98
- 2. 48 9 8
1 48 4
7 8 35
- 3. 19 15.5 19.9 14 14.7 11 16.2 16 13.6 17
- 4. 276 460 396 520 858 580
- 5. 514 258 72 418 157 359 299 30 12
323 927 923 354 324 535 479 735 731
957 493 470 83 345 125 209 236 190
452 728 759 849 344 648 809 976 866
137 387 936 182 554 982 932 185 303

6. 14 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. 628 547 277 846 221 364 378 795 377 51

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. 1524073

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 6 1 7 11 3 4 11 9 3 9 11 9 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 22 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

Primeiros passos programação II-python

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

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

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

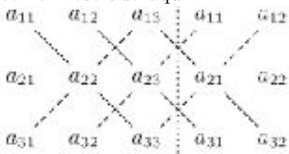
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 quando 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. 20 25 30 36 55 63 69 76 80 81
- 2. 32 5 8
6 28 3
2 2 56
- 3. 17.5 20 12 16 16.6 11.2 13.7 13 15.7 11
- 4. 821 240 258 38 840 192 238 350
- 5. 768 691 783 794 259 681 822 879 670
158 904 150 308 825 326 134 959 576
75 851 662 178 353 329 617 776 893
285 374 746 729 539 132 628 874 106
263 608 664 104 318 49 703 402 513

6. 10 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. 813 756 612 521 928 615 213 224

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. 34267

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

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

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

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 11 22 87 36 37 45 53 64 77
- 2. 3 1 9
1 48 5
9 4 12
- 3. 18 11.2 11 16 13.6 13 15.7 18.7 19.9 13
- 4. 695 630 264 409 714 475 307 447
- 5. 105 609 80 117 616 907 810 784 353
190 95 930 451 996 441 629 681 373
91 577 110 977 233 986 666 722 49
469 667 435 989 932 42 710 947 670
886 632 498 707 67 430 212 539 611

6. 10 11

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

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

7. 948 535 612 369 211 498 757 372 745

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. 1834799

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 8 4 11 1 8 5 1 3 2 9 11 1 6 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 24 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

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

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 18 32 44 52 53 55 87 89 98
- 2. 27 2 2
9 6 2
8 2 10
- 3. 14 16 11 16.8 11.3 20 13.4 14.2 12 13.4
- 4. 328 148 344 205 66 65 290 489
- 5. 319 826 138 29 789 314 920 80 950
291 614 285 351 459 565 353 375 276
215 461 748 907 590 425 163 911 205
167 570 32 705 807 909 391 98 984
115 866 445 55 497 786 854 442 161

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. 153 313 599 582 619 51 826 379 64 644

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. 1595669

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 4 1 4 6 9 9 11 9 8 6 1 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 25 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

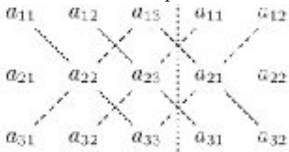
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 1 13 27 30 40 46 54 55 76 98
- 2. 6 5 7
5 20 9
5 5 28
- 3. 19.9 16 17 15.7 12.3 16.2 20 16 14.9 16
- 4. 336 558 992 356 667 55
- 5. 831 253 18 925 555 275 992 428
768 405 507 197 650 626 971 995
292 48 54 529 497 457 702 825
23 915 276 917 424 611 117 832
523 128 203 430 396 868 412 29

6. 14 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. 617 656 596 381 624 156 275 77 123 570

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. 937429

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 9 11 2 8 1 3 11 9 4 11 10 1 4 9

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 26 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

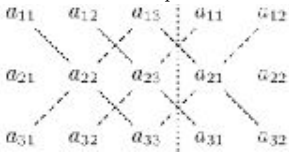
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 com o 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 24 25 26 58 69 79 91 92 94
- 2. 6 4 6
3 63 2
6 6 25
- 3. 16 11.7 12.7 14.2 15 18.6 17.7 11 17 19
- 4. 903 373 652 82 465 56 694 567
- 5. 416 926 465 281 446 319 558 732 700 225
979 846 2 293 841 492 819 510 769 882
383 139 999 421 956 265 915 950 962 895
310 936 805 729 981 8 843 669 913 631
666 513 240 527 177 436 504 803 38 916

6. 10 11

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

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

7. 637 770 953 158 309 891 794 565 113 805 318

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. 1624361

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 11 7 8 10 2 6 4 5 5 3 7 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 27 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

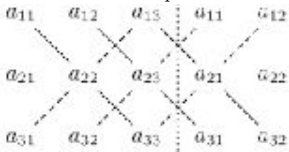
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 11 16 24 28 35 37 43 48 65 98
- 2. 8 6 8
9 9 2
8 4 42
- 3. 19.3 12 13 13 18.9 12 14.2 20 16.5 13.3
- 4. 497 286 2 741 117 827
- 5. 463 460 665 953 870 205 739 894 584
303 280 533 718 852 197 768 813 35
698 38 149 594 967 325 363 451 944
170 923 229 27 647 138 92 347 824
482 327 580 615 689 372 261 844 352

- 6. 11 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. 470 719 884 457 582 454 632 67

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. 1382099

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

- 9. 1 10 8 8 4 2 11 3 4 10 8 11 8 1 9

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 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

Primeiros passos programação II-python

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

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

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

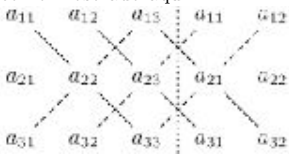
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 29 46 49 52 58 73 74 79 84
- 2. 15 6 7
2 4 7
9 2 20
- 3. 18 17.9 19.1 18 17 12 13.8 20 13.3 16.2
- 4. 315 220 565 188 33 534
- 5. 875 687 133 879 247 845 539 899 909
28 832 857 985 290 781 508 219 778
551 178 799 36 166 686 770 319 204
94 76 702 225 21 301 761 847 412
493 522 863 860 89 334 84 725 794

6. 16 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. 40 949 838 945 238 386 890 456

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. 540805

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 9 10 9 2 8 10 7 1 3 1 9 10 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 29 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

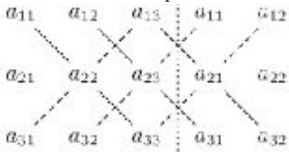
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 1 29 33 37 61 63 70 89 74 99
- 2. 12 4 7
3 21 1
4 4 12
- 3. 13 10.9 12 19.9 12 17 12.8 16 18.3 19.6
- 4. 901 298 980 572 123 470
- 5. 334 984 291 7 257 838 618 937
974 180 2 26 608 225 15 559
742 813 206 819 327 139 445 555
145 932 546 306 441 529 179 851
354 442 44 805 468 732 47 493

6. 16 9

Para os 3 programas a seguir, você deve construir o programa pedido e depois executá-lo com os dados entregues, achando o número em questão e também entregando a listagem do programa que você 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. 841 449 100 16 8 262 429 650 249

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. 530145

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 3 6 3 8 4 9 11 7 3 1 9 5 7 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 30 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

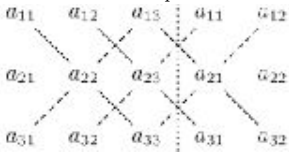
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 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 18 32 35 48 55 60 85 89 95
- 2. 24 4 3
6 63 3
1 7 45
- 3. 12 14 12 17 16.6 11 19 15.4 13 17.1
- 4. 109 218 522 784 825 580 61
- 5. 20 798 696 542 859 113 465 808
176 725 178 945 206 955 65 326
332 933 124 853 734 622 941 862
804 603 446 23 84 731 207 856
229 954 628 551 746 223 62 120

6. 7 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. 492 724 380 142 715 472 804 407 318 528

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. 1379801

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 10 11 5 11 5 8 3 11 9 6 5 1 6 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 31 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

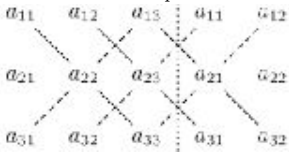
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 11 12 25 38 40 45 51 77 95 98
- 2. 42 7 5
2 10 4
9 7 7
- 3. 15 13 15.3 17.9 12 12.3 18.9 12 19.5 20
- 4. 740 242 115 950 895 53
- 5. 21 313 518 904 85 26 478 62 820 869
668 587 898 431 479 66 334 760 509 247
887 678 866 387 282 392 779 339 578 691
801 780 395 384 995 162 14 686 347 639
428 179 881 746 365 805 93 397 797 91

6. 7 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. 668 843 445 508 774 138 858 951 967 651

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. 1178237

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 2 7 4 3 3 7 8 3 5 3 3 11 5 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 32 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

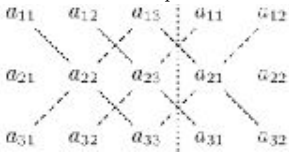
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 15 32 38 45 88 54 61 63 94
- 2. 35 5 4
9 45 1
4 8 30
- 3. 13.9 16.5 18 14.9 18 12 14.3 19.9 13 17
- 4. 779 866 405 783 896 980
- 5. 517 321 81 725 65 35 437 379 842
847 467 147 593 539 601 54 615 693
838 95 682 483 592 537 550 414 689
355 856 963 346 657 429 211 6 655
671 625 870 969 998 950 376 118 733

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. 201 862 331 737 454 497 734 977 4

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. 926155

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 4 10 1 11 8 6 1 3 7 11 7 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 33 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

Primeiros passos programação II-python

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

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

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

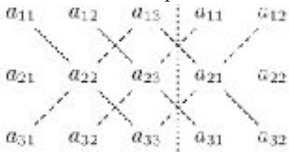
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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 quando 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 38 88 44 50 65 70 75 93 99
- 2. 10 4 6
3 3 3
2 4 5
- 3. 14 17.6 11.4 16 20 16 12 18.7 17 19.1
- 4. 886 563 478 575 329 285
- 5. 399 200 160 9 59 721 283 950
759 900 544 94 691 187 195 692
966 210 166 197 645 552 740 402
683 296 825 51 253 612 599 325
726 768 912 618 658 975 401 727

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. 312 456 558 364 382 979 201 78

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. 98507

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

Responda aqui:

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

observações

- 1. Não esqueça de anexar a listagem dos 3 últimos programas devidamente identificada com o seu nome e com o número 34 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.



Primeiros passos programação II-python

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

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

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

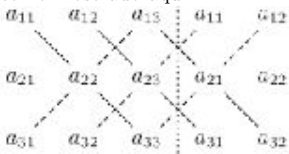
É mais uma tarefa para usar uma sentinela, já que todos os pares possíveis precisarão ser testados e basta que um par não seja verdadeiro, para a lista deixar de estar em ordem. Acompanhe:

```
def orcre(x):
    i=1
    sentinela=True
    while i<len(x) and sentinela==True:
        if x[i]<x[i-1]:
            sentinela=False
        i=i+1
    return sentinela
print(orcre([1, 7, 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. 15 24 29 32 49 63 65 75 76 82
- 2. 16 5 7
7 56 6
5 4 42
- 3. 17.4 11 20 11.5 13 15 18 15.1 10.5 15.8
- 4. 179 142 131 97 597 226 64
- 5. 677 947 871 412 33 973 699 157
29 183 278 360 331 240 508 706
109 140 725 136 480 822 205 252
492 247 842 472 517 830 680 384
646 944 853 636 506 796 353 279

6. 6 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. 705 693 593 646 916 637 638 176 894 370

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. 1475687

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 3 7 5 4 6 2 5 9 7 5 1 6 3 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 35 apos 09/12, 50%
- 2. Em todas as contagens desta folha considere a origem dos índices como UM e não como zero.

