

Ipii20151



Professor 16:44 em 2 de junho de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/06/02/gerar-e-imprimir-as-seguintes-series-em-python/>)

Gerar e imprimir as seguintes séries em python. (M termos ou até N)

- a) 2,4,16,256.....N
- b) 0,1,3,6,10,.....N
- c) 0,N,1,N-1,2,N-2.....N,0
- d) 1,1,2,3,5,8.....N



Daniel Silva Alves Barbosa 17:03 em 2 de junho de 2015

```
1 def probA(n):
2     a = 2
3     while a < n+1:
4         a
5         print a, ", ",
6         a = a*a
7
8 def probB(n):
9     a = 0
10    b = 1
11    while a < n+1:
12        print a, ", ",
13        a = a + b
14        b += 1
15
16 def probC(n):
17     a = 0
18     while a < n+1:
19         print a, " ",
20         print n-a, ", ",
21         a += 1
22
23
24 def probD(n):
25     a = 0
26     b = 1
27     c = 0
28     while a < n:
29         print a, " ",
30         a = b+c
31         c = b
32         b = a
```



Luan Barroso 17:13 em 2 de junho de 2015

ITEM A

```
def seq1(ni,nt):  
    nt2=1  
    print(ni,end="")  
    print(", ",end="")  
    while nt2<nt:  
        ni=ni*ni  
        nt2 +=1  
    print(ni,end="")  
    print(", ",end="")
```

ITEM B

```
def seq2(ni,nt):  
    print(ni,end="")  
    print(", ",end="")  
    nt2=1  
    pos=1  
    while nt2<nt:  
        ni=ni+pos  
        nt2 +=1  
        pos+=1  
    print(ni,end="")  
    print(", ",end="")
```

ITEM C

```
def seq3(ni,nf):  
    print(ni,end="")  
    print(", ",end="")  
    print(nf,end="")  
    print(", ",end="")  
    nf2=nf  
    while nf2!=ni:  
        ni=ni+1  
        nf=nf-1  
    print(ni,end="")  
    print(", ",end="")  
    print(nf,end="")  
    print(", ",end="")
```

ITEM D

```
def seq4(ni,nt):
```

```
def seq4(n):  
    n1=1  
    n2=1  
    nt2=1  
    while nt2<n:  
        print(n1,end="")  
        print(", ",end="")  
        print(n2,end="")  
        print(", ",end="")  
        nt2+=1  
        n1=n1+n2  
        n2=n1+n2
```



Luan Barroso 17:22 em 2 de junho de 2015

ITEM A

```
def seq1(ni,nt):  
    nt2=1  
    print(ni,end="")  
    print(", ",end="")  
    while nt2<nt:  
        ni=ni*ni  
        nt2 +=1  
    print(ni,end="")  
    print(", ",end="")
```

ITEM B

```
def seq2(ni,nt):  
    print(ni,end="")  
    print(", ",end="")  
    nt2=1  
    pos=1  
    while nt2<nt:  
        ni=ni+pos  
        nt2 +=1  
        pos+=1  
    print(ni,end="")  
    print(", ",end="")
```

ITEM C

```
def seq3(ni,nf):  
    print(ni,end="")  
    print(", ",end="")  
    print(nf,end="")  
    print(", ",end="")  
    nf2=nf  
    while nf2!=ni:  
        ni=ni+1  
        nf=nf-1  
    print(ni,end="")  
    print(", ",end="")  
    print(nf,end="")  
    print(", ",end="")
```

ITEM D

```
def seq4(ni,nt):
```

```

def seq4(n1):
    n1=1
    n2=1
    nt2=1
    while nt2<nt:
        print(n1,end="")
        print(", ",end="")
        print(n2,end="")
        print(", ",end="")
        nt2+=1
        n1=n1+n2
        n2=n1+n2

```

[\code]



Professor 16:50 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/2-verificar-se-um-numero-e-par-ou/>)

2. Verificar se um numero é par ou ímpar.



Daniel Silva Alves Barbosa 16:57 em 29 de maio de 2015

```

1  def baskara(a, b, c):
2      res = []
3      d = b*b - a*c*4
4      k = (-b - math.sqrt(d))/(2*a)
5      l = (-b + math.sqrt(d))/(2*a)
6      res.append(k)
7      res.append(l)
8      print res

```



Daniel Silva Alves Barbosa 17:02 em 29 de maio de 2015

```

1  def paridade(x):
2      if x%2 == 0:
3          print "par"
4      else:
5          print "impar"

```

Emanuel Douglas 17:00 em 29 de maio de 2015



```

1  #coding:utf-8
2  import math
3  def calculaEquacao(a,b,c):
4      try:
5          delta = (b**2)+(4*a*c)
6          x = []
7          x.append((-b-(math.sqrt(delta)))/2*a)
8          x.append((-b+(math.sqrt(delta))/2*a)
9      except:
10         pass
11     return x
12
13     print("Entre com os valores da equação : ax^2 + bx + c")
14     a = raw_input("a:")
15     b = raw_input("b:")
16     c = raw_input("c:")
17     a,b,c = int(a),int(b),int(c)
18
19     x = calculaEquacao(a,b,c)
20     print(x)

```



Alvaro Alves 17:12 em 29 de maio de 2015

```

1  def raiz2grau ():
2      a = int(input ("Digite o valor de a: "))
3      b = int(input ("Digite o valor de b: "))
4      c = int(input ("Digite o valor de c: "))
5
6      delta = (b**2) - (4*a*c)
7
8      if (delta < 0):
9          print ("Nao tem raizes reais")
10
11     elif (delta == 0):
12         raiz = -b/2.0*a
13         print ("So tem uma função que e %.2f"%raiz)
14     else:
15         r1 = (-b + delta**(1/2))/2.0*a
16         r2 = (-b - delta**(1/2))/2.0*a
17         print ("A primeira raiz e %.2f e a segunda é %.2f"%(r1,r2))

```



Emanuel Douglas 17:13 em 29 de maio de 2015

```

1  #coding:utf-8
2  def isParOrIpar(a):
3      try:
4          a = int(a)
5          if a%2 == 0:
6              return " par"
7          else:
8              return " impar"
9      except:
10         print "a entrada não é numérica"
11
12     numero = raw_input("informe o numero a ser verificado: ")
13     print str(numero)+" é um número"+isParOrIpar(numero)

```



Emanuel Douglas 17:20 em 29 de maio de 2015

Função de verificar quem é maior :

```

1  #coding:utf-8
2  print("informe 3 numeros: ")
3  a = raw_input("1º >> ")
4  b = raw_input("2º >> ")
5  c = raw_input("3º >> ")
6  a,b,c = int(a),int(b),int(c)
7
8  lista = [a,b,c]
9  print(str(max(lista)))

```



Antônio Matheus 17:26 em 29 de maio de 2015

```

1  def paridade(x):
2      if x%2 == 0:
3          print("é par")
4      else:
5          print("não é par")
6
7  x = int(input("Digite um numero para saber se é par: "))
8
9  paridade(x)

```



jose yuri e augusto feynman 17:26 em 29 de maio de 2015


```
1 import math
2 def equacao2grau():
3     a=int(input("escreva o valor de a"))
4     b=int(input("escreva o valor de b"))
5     c=int(input("escreva o valor de c"))
6     d= b*b -(4.0*a*c)
7     if d<0:
8         print("raiz nao existe!!!!")
9         equacao2grau()
10    else:
11        e=math.sqrt(d)
12        r1=(-b -e)/2.0*a
13        r2=(-b +e)/2.0*a
14        print (r1)
15        print (r2)
16
17
18
19
20 equacao2grau()
```



jose yuri e augusto feynman 17:31 em 29 de maio de 2015

```
def paridade():
n=int(input("digite um numero inteiro:"))
if n%2==0:
print("o numero e par")
else:
print("o numero e impar")
paridade()
```



Luan Barroso 17:33 em 29 de maio de 2015

```

1  import math
2
3
4
5  def raiz():
6      a=int(input("Digite a variável a ( a*x² ):"))
7      b=int(input("Digite a variável a ( b*x ):"))
8      c=int(input("Digite o termo independente (c):"))
9      dt=((b*b)-4*(a*c))
10     if dt<0:
11         print(" Não existem raízes para essa equação ")
12     else:
13         s=dt**(1/2)
14         x1=(-b+s)/(2*a)
15         x2=(-b-s)/(2*a)
16         print(x1)
17         print(x2)
18
19
20 raiz()

```



Professor 16:19 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/exercicios-de-recapitulacao-1-calculas-as-raizes-de/>)

Exercicios de Recapitulação:

1. Calcule as raízes de uma equação do 2o grau.



Daniel Silva Alves Barbosa 16:45 em 29 de maio de 2015

```

def raizes(a, b, c):
    res = []
    for x in range(-100,100):
        if a*x*x + b*x + c == 0:
            res.append(x)
    print res

```



Daniel Silva Alves Barbosa 16:58 em 29 de maio de 2015

```

1  def baskara(a, b, c):
2      res = []
3      d = b*b - a*c*4
4      k = (-b - math.sqrt(d))/(2*a)
5      l = (-b + math.sqrt(d))/(2*a)
6      res.append(k)
7      res.append(l)
8      print res

```



Emanuel Douglas 16:56 em 29 de maio de 2015

```

#coding:utf-8
import math
def calculaEquacao(a,b,c):
try:
delta = (b**2)+(4*a*c)
x = []
x.append((-b-(math.sqrt(delta)))/2*a)
x.append((-b+(math.sqrt(delta)))/2*a)
except:
pass
return x

print("Entre com os valores da equação : ax^2 + bx + c")
a = raw_input("a:")
b = raw_input("b:")
c = raw_input("c:")
a,b,c = int(a),int(b),int(c)

• =calculaEquacao(a,b,c)

print(x)

```



Antônio Matheus 17:21 em 29 de maio de 2015

```
1 import math
2
3 def baskara(a, b, c):
4     d = b**2 - 4*a*c
5
6     if d >= 0:
7
8         x1 = (-b + math.sqrt(d))/2*a
9         print(x1)
10
11         if d > 0:
12             x2 = (-b - math.sqrt(d))/2*a
13             print(x2)
14
15     else:
16         print("raizes complexas (delta negativo)")
17
18 print("Digite a, b e c da equação:")
19
20 a = int(input("a: "))
21 b = int(input("b: "))
22 c = int(input("c: "))
23
24 baskara(a, b, c)
25
26 input()
```



italo 17:32 em 29 de maio de 2015

```

import math
def principal():
    print("Cálculo da equação do segundo grau")
    a = int(input("digite o valor de a: "))
    b = int(input("digite o valor de b: "))
    c = int(input("digite o valor de c: "))
    delta = ((b*b) - (4*a*c))
    if (delta == 0):
        x = (-b/(2*a))
        print("Essa equação tem apenas uma raiz: %x"%x)
    elif(delta <0):
        print("não existe raízes para essa equação")
    else:
        print("Existem duas raízes para essa equação:")
        k = math.sqrt(delta)
        x1 = ((-b + k)/(2*a))
        x2 = ((-b - k)/(2*a))
        print(x1)
        print(x2)

```

[+ Seguir](#)

Seguir “Ipii20151”

Crie um site com
WordPress.com



Professor 16:11 em 29 de maio de 2015

Link Permanente (<https://Ipii20151.wordpress.com/2015/05/29/lembretes-ao-novos-alunos-1-toda-a-materia/>)

Lembretes ao novos alunos:

1. Toda a matéria estará neste Blog. Links para as transparências, vídeo aulas, programas fontes, etc.
2. Faltas são levadas a sério e não são retiradas nem em caso de doença. Vocês terão direito a 2a chamada, mas não à retirada de faltas. A chamada será feita após 15 minutos do início da aula. Se você ficar reprovado por faltas NÃO SOLICITE ABONAR pois será perda de tempo para nós.
4. Você fica reprovado com 25% da quantidade de aulas dadas e não da carga horária, ou seja, normalmente você tem direito a menos de 20h de faltas. Tome cuidado com isso.
5. Ver calendário acadêmico no Depto para detalhes sobre início, fim de aulas e recessos.
6. Avaliações serão feitas através de provas normais. Duas por etapa, mas podem ocorrer trabalhos e exercícios. Alguns exercícios podem ou não valer notas ou pontos adicionais.



Professor 16:05 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/video-aulas-https-www-youtube-com-playlist-listpl51430f6c54953b73/>)

Video Aulas



1/164

logicaPythonAula1



Professor 16:04 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/slides-do-curso-slideshare-id40376487doclogicaprogramacao-python-slides-141016195324/>)

Slides do Curso.

Lógica de Programação com Python

Prof. Ronaldo Ramos

1 of 765



Professor 16:02 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/bem-vindos-ao-curso-de-logica-de-programacao/>)

Bem vindos ao curso de lógica de programação II



Professor 16:00 em 29 de maio de 2015

Link Permanente (<https://lpii20151.wordpress.com/2015/05/29/ola-mundo/>)

Olá, mundo!

Esta é sua primeira publicação. Clique no link Editar para modificar ou excluir, ou então [comece uma nova publicação](#). Se preferir, use esta publicação para informar aos leitores o motivo pelo qual você iniciou este blog e o que planeja fazer com ele.

Continue blogando!

[Crie um website ou blog gratuito no WordPress.com](#). Tema: [P2](#) por [WordPress.com](#).