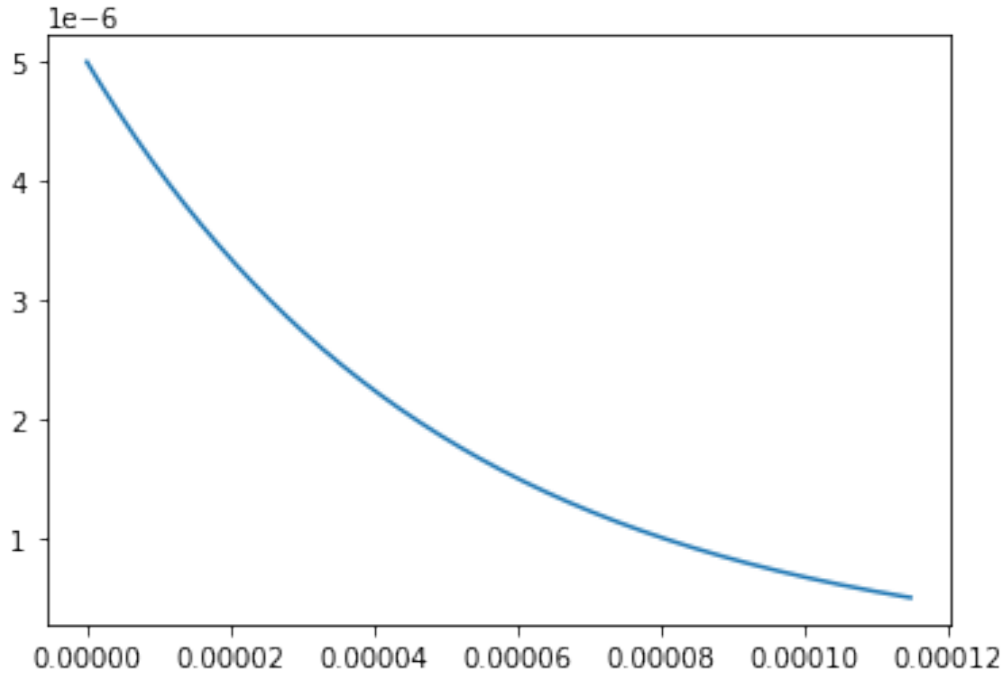
RC problem using Euler method

```
p=[0]
r=[5/1000000]
t=0.
q=5/1000000.
a=float(input("The amount of charge on plate:\n"))
h=(a-q)/1000
for i in range(1000):
    t=t+(h*(-1)*10*5*(1/1000000))/q
    q=q+h
    p.append(t)
    r.append(q)
print(f"At time t={t}s the charge will be q={q}C on the plate.")

import matplotlib.pyplot as plt
plt.plot(p,r)
plt.show()
```

```
The amount of charge on plate:
0.0000005
At time t=0.00011492708877196526s the charge will be
q=5.000000000000411e-07C on the plate.
```

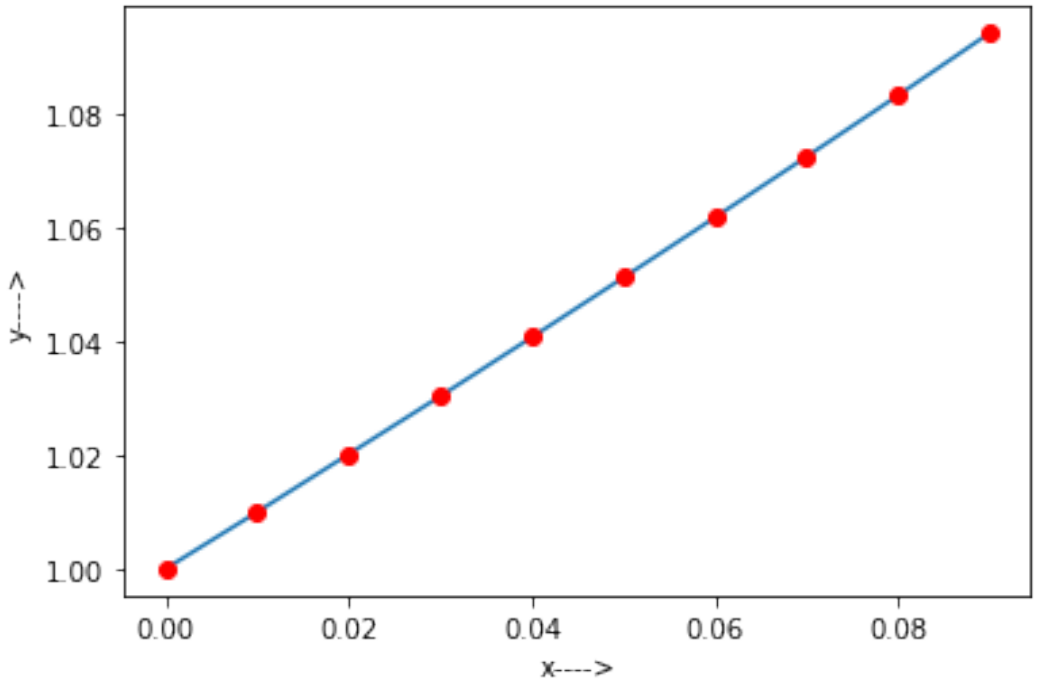



Bgn1 modified ruler

```
def func(x,y):
    y=y+h+((x**2)+y)
    return y
```

```
def func_1(m,n):
    return (m**2)+n

h=0.01
x0=0.
y0=1.

p=[0]
q=[1]
y=func(x0,y0)

while len(p)!=10:
    a=y
    y=y0+((h/2)*(func_1(x0,y0)+func_1(x0+h,y)))
    if int(10000*(a-y)) == 0:
        p.append(x0+h)
        q.append((int(y*10000))/10000)
        # q.append((y//0.0001)/10000)
        x0=x0+h
        y0=y
        y=func(x0,y0)
print(" X=",p,"\n","Y=",q)

import matplotlib.pyplot as plt
plt.plot(p,q)
plt.plot(p,q,"ro")
plt.xlabel("x---->")
plt.ylabel("y---->")
plt.show()
```

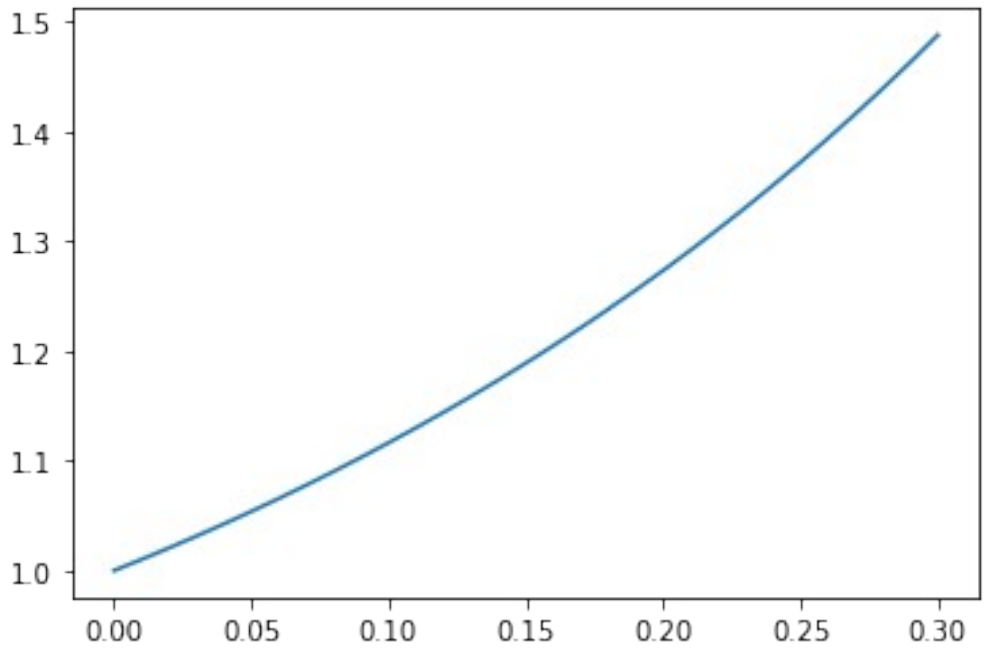
```
 X= [0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.060000000000000005, 0.07,
0.08, 0.09]
 Y= [1, 1.01, 1.0202, 1.0304, 1.0408, 1.0513, 1.0619, 1.0726, 1.0834,
1.0944]
```




Bgn2

```
def func(m,n):
  return m+(n*n)
p=[0]
q=[1]
h=0.01
x0=0.
y0=1.
y=y0
for i in range(30):
  k1=h*func(x0,y0)
  k2=h*func(x0+(h/2),y0+(k1/2))
  k3=h*func(x0+(h/2),y0+(k2/2))
  k4=h*func(x0+h,y0+k3)
  y=y+(1/6)*(k1+(2*k2)+(2*k3)+k4)
  x0=x0+h
  y0=y
  p.append(x0)
  q.append(y)
  print(f"The value of y{i+1}={y}")

import matplotlib.pyplot as plt
plt.plot(p,q)
```

```
The value of y1=1.0101513476535966
The value of y2=1.0206108983952318
The value of y3=1.0313871864021218
The value of y4=1.0424891257916191
The value of y5=1.0539260325256656
The value of y6=1.0657076478168446
The value of y7=1.0778441631580908
The value of y8=1.090346247109606
The value of y9=1.1032250739892417
The value of y10=1.1164923546267154
The value of y11=1.1301603693576572
The value of y12=1.1442420034508738
The value of y13=1.1587507851815397
The value of y14=1.173700926784564
The value of y15=1.1891073685463915
The value of y16=1.204985826320324
The value of y17=1.2213528427804288
The value of y18=1.2382258427627042
The value of y19=1.2556231930798318
The value of y20=1.2735642672381642
The value of y21=1.2920695155331883
The value of y22=1.3111605410533314
The value of y23=1.330860182182488
The value of y24=1.3511926022600478
The value of y25=1.372183387134645
The value of y26=1.3938596514356902
The value of y27=1.4162501544865531
The value of y28=1.4393854268968704
The value of y29=1.4632979090010367
The value of y30=1.488022102457996

[<matplotlib.lines.Line2D at 0x7ff2ee3110d0>]
```

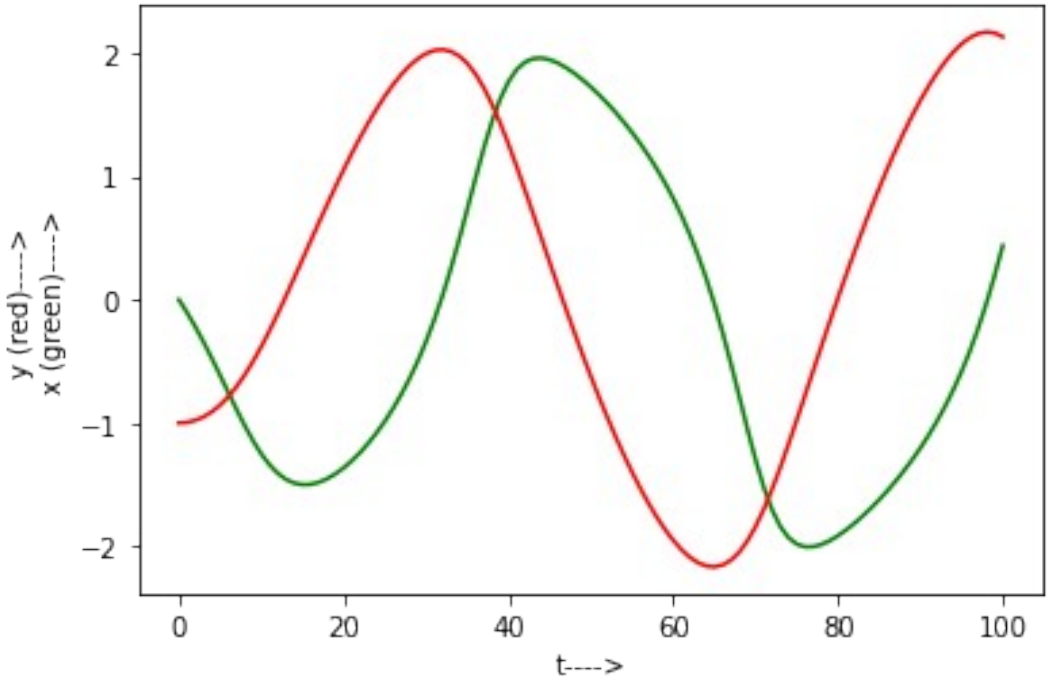



Bgn2a: #Coupled differential eq. using RK4

```
def func(m,n):
  return n+m-(m*m*m)/3

def func_1(r,s):
  return -r

p=[0]
q=[-1]
t=[0]
h=0.1
x0=0.
y0=-1.
y=y0
for i in range(100):
  k1x=h*func(x0,y0)
  k1y=h*func_1(x0,y0)
  k2x=h*func(x0+(k1x/2),y0+(k1y/2))
  k2y=h*func_1(x0+(k1x/2),y0+(k1y/2))
  k3x=h*func(x0+(k2x/2),y0+(k2y/2))
  k3y=h*func_1(x0+(k2x/2),y0+(k2y/2))
  k4x=h*func(x0+k3x,y0+k3y)
  k4y=h*func_1(x0+k3x,y0+k3y)
  kx=(1/6)*(k1x+(2*k2x)+(2*k3x)+k4x)
  ky=(1/6)*(k1y+(2*k2y)+(2*k3y)+k4y)
  x0=x0+kx
  y0=y0+ky
  p.append(x0)
  q.append(y0)
```

```
  t.append(i+1)

print(f"The value of x{i+1}={kx} y{i+1}={ky}")

import matplotlib.pyplot as plt
plt.plot(t,p,color="g")
plt.plot(t,q,color="r")
plt.xlabel("t---->")
plt.ylabel("x---->")
plt.ylabel("y (red)---->\nx (green)---->")
plt.show()
```

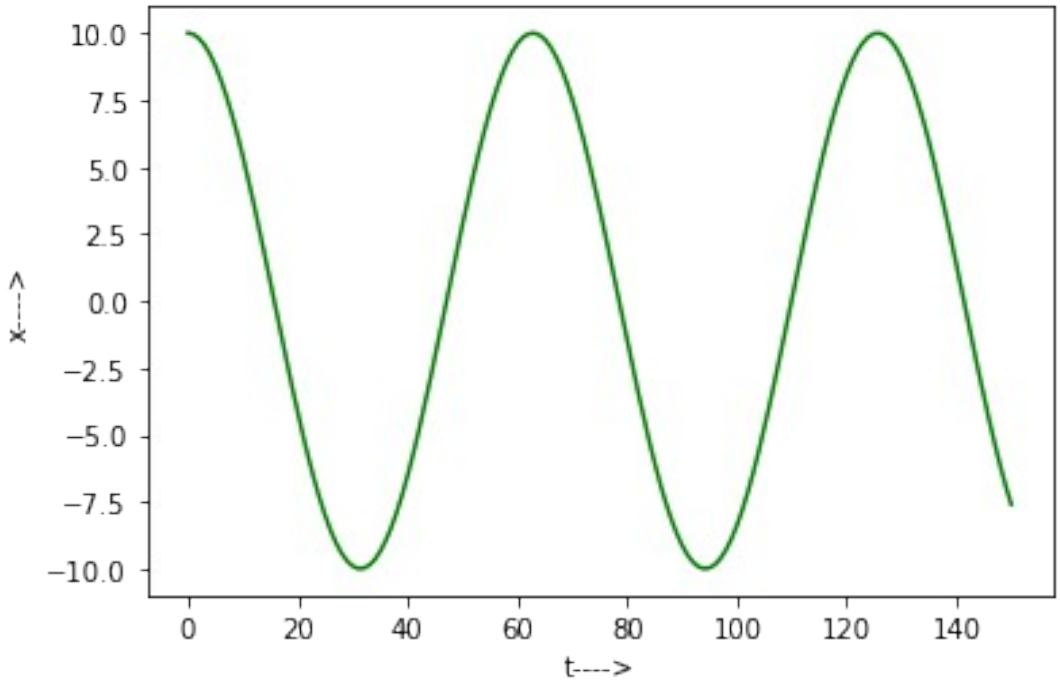
```
The value of x100=0.2453701421483732 y100=-0.03150466290938982
```




Bgn2b: Damped harmonic oscillator (Second to two first order coupled differential eq./RK4

```
u=float(input("Enter the value of u:\n"))
m=1
k=1

def func(m,n):
    return n

def func_1(r,s):
    return -(u/m)*s-(k/m)*r

p=[10]
```

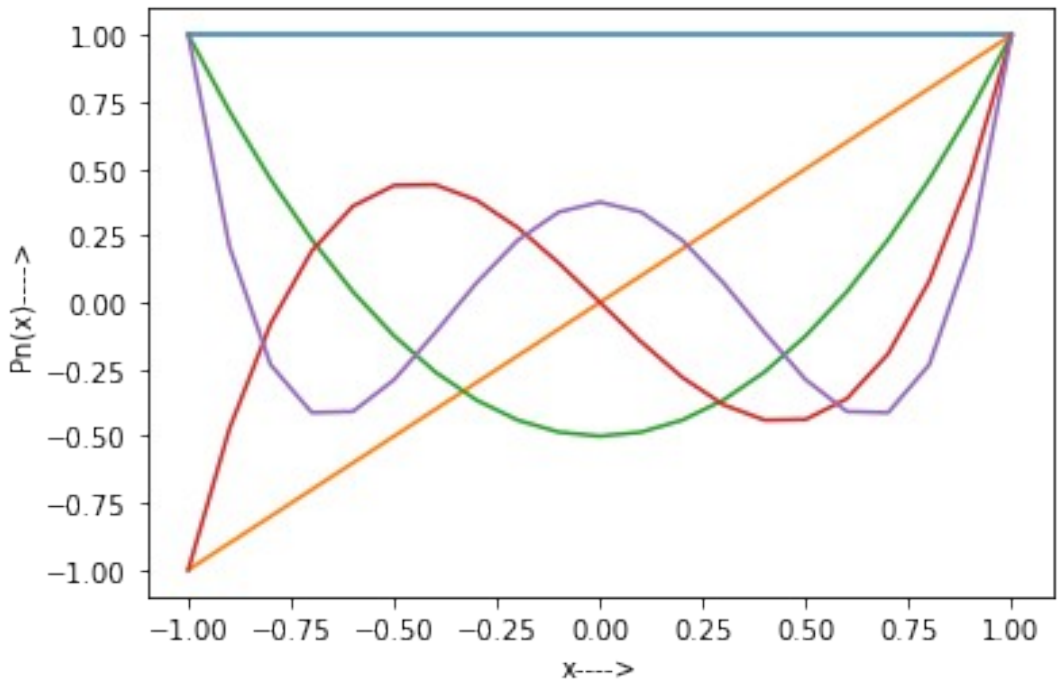
```
t=[0]
h=0.1
x0=10
y0=0

for i in range(150):
    k1x=h*func(x0,y0)
    k1y=h*func_1(x0,y0)
    k2x=h*func(x0+(k1x/2),y0+(k1y/2))
    k2y=h*func_1(x0+(k1x/2),y0+(k1y/2))
    k3x=h*func(x0+(k2x/2),y0+(k2y/2))
    k3y=h*func_1(x0+(k2x/2),y0+(k2y/2))
    k4x=h*func(x0+k3x,y0+k3y)
    k4y=h*func_1(x0+k3x,y0+k3y)
    kx=(1/6)*(k1x+(2*k2x)+(2*k3x)+k4x)
    ky=(1/6)*(k1y+(2*k2y)+(2*k3y)+k4y)
    x0=x0+kx
    y0=y0+ky
    p.append(x0)
    t.append(i+1)
print(f"The value of x{i+1}={kx} y{i+1}={ky}")

import matplotlib.pyplot as plt
plt.plot(t,p,color="g")
plt.xlabel("t---->")
plt.ylabel("x---->")
plt.show()

Enter the value of u:
0
The value of x150=-0.6871650000612717 y150=0.7259252558455385
```




Bgn3a

```
import matplotlib.pyplot as plt
def func(n,x):
    if n==0:
        return 1
    elif n==1:
        return x
    else:
         return (((2*(n-1)+1)/((n-1)+1))*x*func(n-1,x))-(((n-1)/((n-
1)+1))*func(n-2,x))

x=-1.1
h=0.1

for i in range(5):
    p=[]
    q=[]
    for j in range(21):
        x=x+h
        y=func(i,x)
        p.append(x)
        q.append(y)
    plt.plot(p,q)
    x=-1.1

plt.xlabel("x---->")
plt.ylabel("Pn(x)---->")
```

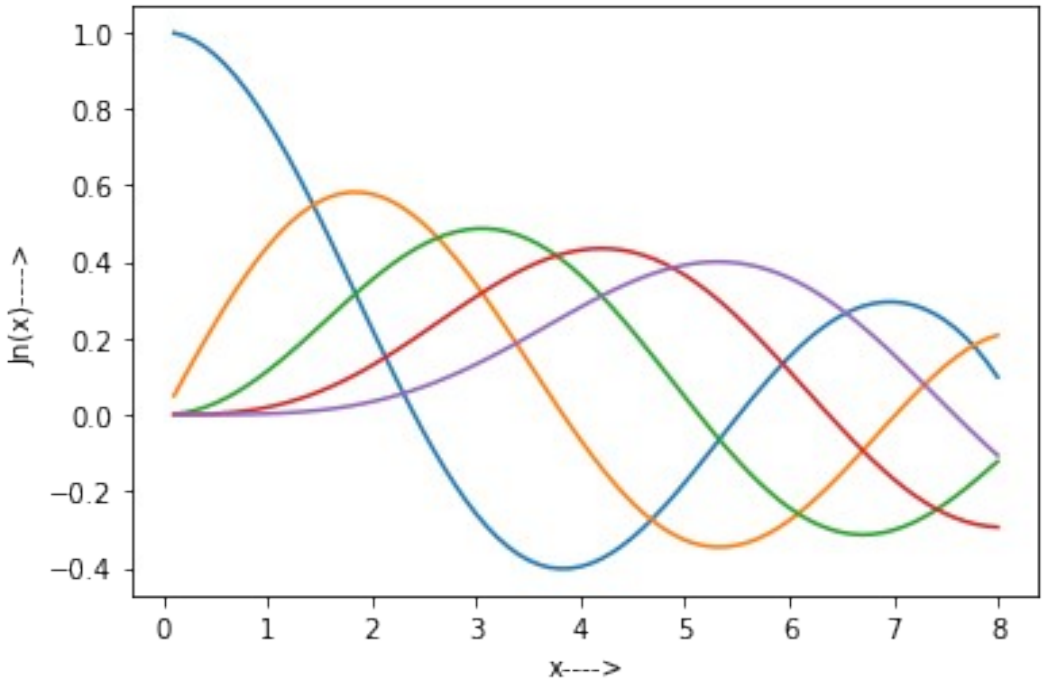
```
plt.show()
```




Bgn3: Generating and plotting of Bessel's Polynomial

```
import matplotlib.pyplot as plt
import math

def func(x,r,n):
  return (((-
1)**(r))*((x/2)**(n+(2*r))))/((math.factorial(r))*(math.gamma(r+n+1)))

x=0.
z=0.
h=0.1
for k in range(5):
  p=[]
  q=[]
  for i in range(80):
    x=x+h
    for j in range(10):
      z=z+func(x,j,k)
    q.append(z)
    p.append(x)
    z=0
  x=0
  plt.plot(p,q)

plt.xlabel("x---->")
```

```
plt.ylabel("Jn(x)---->")
plt.show()
```




Bgn4: extra

```
import numpy as np
a=np.array([[200,-100],[100,-150]])
b=np.array([[12],[0]])
c=np.linalg.inv(a)
d=np.dot(c,b)
print("The matrix A:\n",a)
print("\nThe matrix B:\n",b)
print("\nThe inverse of matrix A:\n",c)
print("\nThe matrix X:\n",d)
print(f"\nI1={d[0]}\nI2={d[1]}")
print(f"\nTherefore, the current through RL:\nI2={d[1]}")
```

```
The matrix A:
 [[ 200 -100]
 [ 100 -150]]

The matrix B:
 [[12]
 [ 0]]

The inverse of matrix A:
 [[ 0.0075 -0.005 ]
 [ 0.005  -0.01  ]]
```

```
The matrix X:
 [[0.09]
 [0.06]]

I1=[0.09]
I2=[0.06]

Therefore, the current through RL:
I2=[0.06]
```

Bgn4a:

```
import numpy as np
a=np.array([[8,-1,-3],[-3,-10,15],[-1,13,-10]])
b=np.array([[24],[-10],[12]])
c=np.linalg.inv(a)
d=np.dot(c,b)
print("The matrix A:\n",a)
print("\nThe matrix B:\n",b)
print("\nThe inverse of matrix A:\n",c)
print("\nThe matrix X:\n",d)
print(f"\nI1={d[0]}\nI2={d[1]}\nI3={d[2]}")
print(f"\nTherefore, the current through 4 ohm resistor:\nI1={d[0]}")
```

```
The matrix A:
 [[  8  -1  -3]
 [ -3 -10  15]
 [ -1  13 -10]]

The matrix B:
 [[ 24]
 [-10]
 [ 12]]

The inverse of matrix A:
 [[0.16725352 0.08626761 0.07922535]
 [0.07922535 0.14612676 0.19542254]
 [0.08626761 0.18133803 0.14612676]]

The matrix X:
 [[4.10211268]
 [2.78521127]
 [2.01056338]]

I1=[4.10211268]
I2=[2.78521127]
I3=[2.01056338]

Therefore, the current through 4 ohm resistor:
I1=[4.10211268]
```

Bgn:5 extra

```
a=[34,45,59,73,98]
n=len(a)-1
rr=[]
rr.append(a)
for i in range(n):
    b=[]
    for j in range(n):
        m=a[j+1]-a[j]
        b.append(m)
    rr.append(b)
    n=n-1
    a=b
print(rr)
```

```
[[34, 45, 59, 73, 98], [11, 14, 14, 25], [3, 0, 11], [-3, 11], [14]]
```

Bgn:6 extra

```
a=[1.6596,1.6698,1.6804,1.6912,1.7024,1.7139]
n=len(a)-1
rr=[]
rr.append(a)
for i in range(n):
    b=[]
    for j in range(n):
        m=a[j+1]-a[j]
        b.append(m)
    rr.append(b)
    n=n-1
    a=b
print(rr)

import math
h=0.02
x0=0.20
x=float(input("Enter the value of x:\n"))  #x=0.23 or x=0.29
p=(x-x0)/h
y=1.6596
def func(r):
    if r>p:
        return 1
    else:
        z=r*func(r+1)
        return z

for k in range(len(rr)-1):
    y=y+(((rr[k+1][0])*func(p-k))/math.factorial(k+1))
print(f"F({x})={y}")
```

```
[[1.6596, 1.6698, 1.6804, 1.6912, 1.7024, 1.7139],
[0.010199999999999987, 0.010599999999999943, 0.010800000000000143,
0.011199999999999877, 0.011500000000000066], [0.00039999999999995595,
0.00020000000000020002, 0.00039999999999997339, 0.00030000000000000189],
[-0.00019999999999975593, 0.00019999999999953388, -9.99999999995449e-
05], [0.00039999999999992898, -0.00029999999999990788], [-
0.00069999999999983686]]
Enter the value of x:
0.23
F(0.23)=1.6750800781250001
```

Bgn7

```
def func(x):
  y= 1/(1+x)
  return y
b=1
a=0
p=[]
h=(b-a)/10
x=0
for i in range(11):
  y=func(x)
  p.append(y)
  x=x+h
print(p)
m=0
for j in range(9):
  n=p[0]+p[10]
  m=m+(2*(p[j+1]))
rlt=(m+n)*(h/2)
print(rlt)

r=0
for k in range(1,5):
  r=r+(2*p[2*k])
s=0
for l in range(5):
  s=s+4*p[(2*l+1)]
print((r+s+n)*(h/3))
```

```
[1.0, 0.9090909090909091, 0.8333333333333334, 0.7692307692307692,
0.7142857142857143, 0.6666666666666666, 0.625, 0.5882352941176471,
0.5555555555555556, 0.5263157894736842, 0.5]
0.6937714031754278
0.6931502306889303
```

bgnextra1

```
def func(x):
  y= 1/(1+x)
```

```
  return y
b=1
a=0
p=[]
h=(b-a)/10
x=0
for i in range(11):
  y=func(x)
  p.append(y)
  x=x+h
print(p)
m=0
for j in range(1,10):
  if j%3==0:
    m=m+(2*p[j])
  else:
    m=m+(3*p[j])
print((m+p[0]+p[10])*((3/8)*h))
```

```
[1.0, 0.9090909090909091, 0.8333333333333334, 0.7692307692307692,
0.7142857142857143, 0.6666666666666666, 0.625, 0.5882352941176471,
0.5555555555555556, 0.5263157894736842, 0.5]
0.6803473326209394
```

Bgn8

```
p=[]
for i in range(3):
  q=[]
  for j in range(4):
    a=input(f"Enter the value at a({i}{j}) = ")
    a=int(a)
    q.append(a)
  p.append(q)
print("\nList:",p)

def func(y,z):
  m=(1/p[0][0])*((p[0][3])-((p[0][1])*y)-((p[0][2])*z))
  return m

def func_1(x,z):
  n=(1/p[1][1])*((p[1][3])-((p[1][0])*x)-((p[1][2])*z))
  return n

def func_2(x,y):
  o=(1/p[2][2])*((p[2][3])-((p[2][0])*x)-((p[2][1])*y))
  return o

r=[0]
s=[0]
```

```
t=[0]

x=0
y=0
z=0
for i in range(20):
  a=func(y,z)
  x=a
  b=func_1(x,z)
  y=b
  c=func_2(x,y)
  z=c
  r.append(a)
  s.append(b)
  t.append(c)

print("\nx = ",r)
print("\ny = ",s)
print("\nz = ",t)
```

```
Enter the value at a(00) = 2
Enter the value at a(01) = 1
Enter the value at a(02) = 1
Enter the value at a(03) = 5
Enter the value at a(10) = 3
Enter the value at a(11) = 5
Enter the value at a(12) = 2
Enter the value at a(13) = 15
Enter the value at a(20) = 2
Enter the value at a(21) = 1
Enter the value at a(22) = 4
Enter the value at a(23) = 8

List: [[2, 1, 1, 5], [3, 5, 2, 15], [2, 1, 4, 8]]

x =  [0, 2.5, 1.5625, 1.1734375, 1.0434765625, 1.0076552734375,
1.0000541259765625, 0.9992655950927734, 0.9995723045806884,
0.9998275156246184, 0.9999435910260867, 0.999984813277538,
0.9999969085319755, 0.9999997410781319, 1.0000001602667283,
1.0000001180455556, 1.0000000518495216, 1.0000000180089192,
1.0000000051701765, 1.0000000011719594, 1.000000000156859]

y =  [0, 1.5, 1.9125, 1.9996875000000003, 2.0085703125000003,
2.0049591796875, 2.0019944970703127, 2.0006509178466803,
2.00017482805481, 2.000035434346848, 2.0000028919439568,
1.9999981194330905, 1.9999986294796317, 1.9999994000074794,
1.9999997920563375, 1.9999999404316462, 1.999999986542563,
1.999999998218809, 2.0000000003215592, 2.0000000003630154,
2.0000000001765783]
```

```
z =  [0, 0.375, 0.740625, 0.913359375, 0.9761191406249999,
0.994932568359375, 0.9994743127441406, 1.000204472991943,
1.0001701406959533, 1.0000773836009789, 1.0000274815009673,
1.0000080635029582, 1.0000018883641044, 1.0000002794590641,
0.9999999718525514, 0.9999999558693107, 0.9999999774395985,
0.9999999914408382, 0.999999997334522, 0.9999999993232664,
0.9999999998774259]

print(3%4)
print(6%3)

3
0
```

bgnextra

```
p=[]
for i in range(3):
  q=[]
  for j in range(4):
    a=input()
    a=int(a)
    q.append(a)
  p.append(q)
print(p)

def func(y,z):
  m=(1/p[0][0])*((p[0][3])-((p[0][1])*y)-((p[0][2])*z))
  return m

def func_1(x,z):
  n=(1/p[1][1])*((p[1][3])-((p[1][0])*x)-((p[1][2])*z))
  return n

def func_2(x,y):
  o=(1/p[2][2])*((p[2][3])-((p[2][0])*x)-((p[2][1])*y))
  return o

r=[0]
s=[0]
t=[0]

x=0
y=0
z=0
for i in range(20):
  a=func(y,z)
  x=a
  b=func_1(x,z)
  y=b
```

```
  c=func_2(x,y)
  z=c
  r.append(a)
  s.append(b)
  t.append(c)
print(r)
print(s)
print(t)
```

```
3
-2
0
10
-5
10
-2
0
0
-8
108
0
[[3, -2, 0, 10], [-5, 10, -2, 0], [0, -8, 108, 0]]
[0, 3.333333333333333, 4.444444444444444, 4.831275720164609,
4.965950312452369, 5.01283702235996, 5.0291605435870474,
5.034843547273515, 5.0368220744828776, 5.0375108950668785,
5.037750706677604, 5.037834196645783, 5.037863263523594,
5.037873383103276, 5.037876906216202, 5.037878132781444,
5.03787855980786, 5.037878708476317, 5.037878760234965,
5.037878778254642, 5.037878784528159]
[0, 1.6666666666666665, 2.2469135802469133, 2.448925468678555,
2.5192555335399414, 2.543740815380572, 2.552265320910273,
2.555233111724317, 2.556266342600318, 2.5566260600164075,
2.5567512949686755, 2.5567948952853907, 2.556810074654914,
2.5568153593243035, 2.556817199172165, 2.556817839711791,
2.556818062714475, 2.5568181403524473, 2.556818167381963,
2.5568181767922393, 2.5568181800684093]
[0, 0.12345679012345677, 0.16643804298125284, 0.18140188656878184,
0.1866115210029586, 0.18842524558374607, 0.18905669043779796,
0.18927652679439386, 0.18935306241483835, 0.1893797081493635,
0.18938898481249447, 0.1893922144655845, 0.18939333886332693,
0.18939373032031875, 0.18939386660534555, 0.18939391405272524,
0.1893939305714426, 0.18939393632240348, 0.18939393832458987,
0.18939393902164733, 0.1893939392643266]
```

Bgn9

```
a=[0.1003,0.1511,0.2027,0.2553,0.3093]
n=len(a)-1
rr=[]
rr.append(a)
```

```
for i in range(n):
    b=[]
    for j in range(n):
        m=a[j+1]-a[j]
        b.append(m)
    rr.append(b)
    n=n-1
    a=b
print(rr)

import math
h=0.05
x0=0.10
x=float(input("Enter the value of x:\n"))  #x=0.12
p=(x-x0)/h
y=0.1003
def func(r):
    if r>p:
        return 1
    else:
        z=r*func(r+1)
        return z

for k in range(len(rr)-1):
    y=y+(((rr[k+1][0])*func(p-k))/math.factorial(k+1))
print(f"tan({x})={y}")
```

```
[[0.1003, 0.1511, 0.2027, 0.2553, 0.3093], [0.05080000000000001,
0.05159999999999998, 0.05260000000000036, 0.05399999999999999],
[0.0007999999999999674, 0.0010000000000000564, 0.0013999999999999568],
[0.00020000000000000089, 0.0003999999999990044],
[0.00019999999999981144]]
Enter the value of x:
0.12
tan(0.12)=0.12053520000000004
```

Bgn9a

```
a=[0.2588190,0.3420201,0.4226185,0.5,0.5735764,0.6427876]
n=len(a)-1
rr=[]
rr.append(a)
for i in range(n):
    b=[]
    for j in range(n):
        m=a[j+1]-a[j]
        b.append(m)
    rr.append(b)
    n=n-1
    a=b
print(rr)
```

```
import math
h=5
xn=40
x=float(input("Enter the value of x:\n"))  #x=0.12
p=(x-xn)/h
y=0.6427876
def func(r):
    if r<p:
        return 1
    else:
        z=r*func(r-1)
        return z

for k in range(len(rr)-1):
    y=y+(((rr[k+1][-1])*func(p+k))/math.factorial(k+1))
print(f"sin({x})={y}")
```

```
[[0.258819, 0.3420201, 0.4226185, 0.5, 0.5735764, 0.6427876],
[0.08320109999999997, 0.08059840000000001, 0.07738149999999999,
0.07357639999999999, 0.06921120000000003], [-0.002602699999999958, -
0.0032169000000000225, -0.0038051000000000057, -0.004365199999999958],
[-0.0006142000000000647, -0.0005881999999999832, -
0.0005600999999999523], [2.6000000000081513e-05, 2.8100000000030878e-
05], [2.0999999999493646e-06]]
Enter the value of x:
38
sin(38.0)=0.6156615585408
```

Bgn 10

```
f=[0,2,4,6,8,10,12]
g=[0,22,30,27,18,7,0]
for i in range(len(g)):
    g[i]=(50/3)*g[i]
print("Velocity(m/min.)=",g)

def func(u,t,z):
    v=(u*t)+z
    return v

res=0
for rr in range((len(g)-1)):
    b=f[rr+1]
    a=f[rr]
    p=[]
    h=(b-a)/10
    t=f[rr]
    u=(g[rr+1]-g[rr])/(f[rr+1]-f[rr])
    z=(-u*f[rr])+g[rr]
```

```
    for i in range(11):
        y=func(u,t,z)
        p.append(y)
        t=t+h

    n=p[0]+p[10]
    r=0
    for k in range(1,5):
      r=r+(2*p[2*k])
    s=0
    for l in range(5):
      s=s+4*p[(2*l+1)]
    res=res+((r+s+n)*(h/3))
    print(f"Displacement of car, x{rr+1}={res}")
```

```
Velocity(m/min.)= [0.0, 366.6666666666667, 500.00000000000006,
450.00000000000006, 300.0, 116.66666666666667, 0.0]
Displacement of car, x1=366.66666666666674
Displacement of car, x2=1233.3333333333335
Displacement of car, x3=2183.3333333333335
Displacement of car, x4=2933.3333333333335
Displacement of car, x5=3350.000000000001
Displacement of car, x6=3466.666666666668
```