1	a	Python Program to solve $\frac{dy}{dx} = 2y + 3e^x, y(0) = 0$ at $x = 0.1$ using Taylor's series method considering terms up to 4th degree.

```
from sympy import *
x,y=symbols('x,y')
f=2*y+3*exp(x)
x0=float(input('Enter the initial value for x : '))
y0=float(input('Enter the initial value for y : '))
n=int(input('Enter the number of terms required in the series : '))
print('y1 = %0.4f'%f.subs({x:x0 , y:y0}))
series=y0+(x-x0)*(f.subs({x:x0 , y:y0}))
dy=f
for i in range (2,n):
    dy=diff(dy,x)+diff(dy,y)*f
    dy0=dy.subs({x:x0,y:y0})
    print(f'y{i} = %0.4f'%dy0)
    series=series+((((x-x0)**i)*dy0)/factorial(i))
display(series)
xvalue=float(input('Enter the value of x at which we have to find y :
print(f'y({xvalue}) = %0.4f'%series.subs({x:xvalue , y:y0}))
```

OUT PUT

```
Enter the initial value for x : 0
Enter the initial value for y : 0
Enter the number of terms required in the series : 5
y1 = 3.0000
y2 = 9.0000
y3 = 21.0000
y4 = 45.0000
```

$$\frac{15x^4}{8} + \frac{7x^3}{2} + \frac{9x^2}{2} + 3x$$

```
Enter the value of x at which we have to find y : 0.1
y(0.1) = 0.3487
```

2	a	Python program to solve $\frac{dy}{dx} = x + \sin(y), y(0) = 1$ at $x = 0.4$ using Modified Euler's method in 2 stages taking $h = 0.2$.

```python
from sympy import *
x,y=symbols('x,y')
f=x+sin(y)
x0=float(input('Enter the initial value for x : '))
y0=float(input('Enter the initial value for y : '))
h=float(input('Enter the step lenght h : '))
n=int(input('Enter the number of iteration required in Modified Eulers M
m=int(input('Enter the total number of values of x at which y should be
for i in range (1,m+1):
    yE=y0+h*f.subs({x:x0 , y:y0})
    x1=x0+h
    yME=yE
    print('\nFrom Eulers Method : y = %0.4f'%yE)
    print('\nFrom Modified Eulers Method')
    for j in range (1,n+1):
        yME=y0+(h/2)*(f.subs({x:x0 , y:y0})+f.subs({x:x1 , y:yME}))
        print(f'{j} - Iteration : y = %0.4f'%yME)
    x0=x1
    y0=yME
```

OUT PUT

```
Enter the initial value for x : 0
Enter the initial value for y : 1
Enter the step lenght h : 0.2
Enter the number of iteration required in Modified Eulers Method : 2
Enter the total number of values of x at which y should be determainec
2

From Eulers Method : y = 1.1683

From Modified Eulers Method
1 - Iteration : y = 1.1962
2 - Iteration : y = 1.1972

From Eulers Method : y = 1.4234

From Modified Eulers Method
1 - Iteration : y = 1.4492
2 - Iteration : y = 1.4496
```

3	a	Python program to solve $\frac{dy}{dx}=2y+3e^{x}, y(0)=0$ at $x=0.1$ using the Runge-Kutta method by taking $h=0.1$.

```python
from sympy import *
x,y=symbols('x,y')
f=2*y+3*exp(x)
x0=float(input('Enter the initial value of x : '))
y0=float(input('Enter the initial value of y : '))
h=float(input('Enter the value for step length h = '))

k1=h*f.subs({x:x0 , y:y0})
k2=h*f.subs({x:x0+(h/2) , y:y0+(k1/2)})
k3=h*f.subs({x:x0+(h/2) , y:y0+(k2/2)})
k4=h*f.subs({x:x0+h , y:y0+k3})
solution=y0+(1/6)*(k1+(2*k2)+(2*k3)+k4)

print('\nk1 = %0.4f'%k1,'\tk2 = %0.4f'%k2,'\tk3 = %0.4f'%k3,'\tk4 = %0.4f'%
print(f'y({x0+h})=%0.4f'%solution)
```

<u>OUT PUT</u>

```
Enter the initial value of x : 0
Enter the initial value of y : 0
Enter the value for step length h = 0.1

k1 = 0.3000     k2 = 0.3454     k3 = 0.3499     k4 = 0.4015
y(0.1)=0.3487
```

4 a Python program to compute $y(0.4)$ using Milne's method (apply corrector formula thrice), given $\frac{dy}{dx} = 2e^x + y$ with

x	0	0.1	0.2	0.3
y	2.4	2.473	3.129	4.059

```
from sympy import *
x,y=symbols('x,y')
f=2*exp(x)+y
h=float(input('Enter the value for step length h = '))
print('Enter the values for x and y ')
x0=float(input('x0 = '))
x1=x0+h
x2=x1+h
x3=x2+h
x4=x3+h
y0=float(input('y0 : '))
y1=float(input('y1 : '))
y2=float(input('y2 : '))
y3=float(input('y3 : '))
f1=f.subs({x:x1 , y:y1})
f2=f.subs({x:x2 , y:y2})
f3=f.subs({x:x3 , y:y3})
y4p=y0+((4*h)/3)*(2*f1-f2+2*f3)
print(f'\nFrom Milnes Predictor formula y({x4})=%0.4f'%y4p)
f4=f.subs({x:x4 , y:y4p})
yc=0
i=1
print('\nFrom Milnes Corrector formula')
while i<=3:
    y4c=y2+(h/3)*(f2+4*f3+f4)
    dif=abs(y4c-yc)
    yc=y4c
    print(f'{i} - Iteration : y({x4})=%0.4f'%y4c)
    f4=f.subs({x:x4 , y:y4c})
    i+=1
```

OUT PUT

```
Enter the value for step length h = 0.1
Enter the values for x and y
x0 = 0
y0 : 2.4
y1 : 2.473
y2 : 3.129
y3 : 4.059

From Milnes Predictor formula y(0.4)=4.7083

From Milnes Corrector formula
1 - Iteration : y(0.4)=4.4723
2 - Iteration : y(0.4)=4.4644
3 - Iteration : y(0.4)=4.4642
```

5 a Python Program to find the normal vector to the surface $\phi = x^2 y + y^2 z + z^2 x$ at $(1,1,1)$.

```
from sympy . physics . vector import *
from sympy import *
x,y,z=symbols('x,y,z')
v= ReferenceFrame ('v')
φ=x**2*y+y**2*z+z**2*x
print ("\n Gradient of φ is")
Δφ=display(Derivative(φ,x)*v.x+Derivative(φ,y)*v.y+Derivative(φ,z)*v
gradφ=diff(φ,x)*v.x+diff(φ,y)*v.y+diff(φ,z)*v.z
display(gradφ)
NV=gradφ.subs({x:1,y:1,z:1})
print('Normal vector to the suface is ')
display(NV)
```

<u>OUT PUT</u>

```
 Gradient of φ is
```

$$\frac{\partial}{\partial x}\left(x^2 y + x z^2 + y^2 z\right)\hat{\mathbf{v}}_{\mathbf{x}} + \frac{\partial}{\partial y}\left(x^2 y + x z^2 + y^2 z\right)\hat{\mathbf{v}}_{\mathbf{y}} + \frac{\partial}{\partial z}\left(x^2 y + x z^2 + y^2 z\right)\hat{\mathbf{v}}_{\mathbf{z}}$$

$$(2xy + z^2)\hat{\mathbf{v}}_{\mathbf{x}} + (x^2 + 2yz)\hat{\mathbf{v}}_{\mathbf{y}} + (2xz + y^2)\hat{\mathbf{v}}_{\mathbf{z}}$$

```
Normal vector to the suface is
```

$$3\hat{\mathbf{v}}_{\mathbf{x}} + 3\hat{\mathbf{v}}_{\mathbf{y}} + 3\hat{\mathbf{v}}_{\mathbf{z}}$$

6 a Python program to verify whether the vector $\left(-x^2 + yz\right)\hat{i} + \left(-z^2 x + 4y\right)\hat{j} + \left(2xz - 4z\right)\hat{k}$ is solenidal.

```python
from sympy . physics . vector import *
from sympy import *
x,y,z=symbols('x,y,z')
v= ReferenceFrame ('v')
f1=-(x**2)+y*z
f2=-(z**2)+4*y
f3=2*x*z-4*z
f=f1*v.x+f2*v.y+f3*v.z
print ("\n Divergence of the vector f is ")
display(Derivative(f1,x)+Derivative(f2,y)+Derivative(f3,z))
divf=diff(f1,x)+diff(f2,y)+diff(f3,z)
display(divf)
if divf==0:
    print('The vector f is solenoidal')
else:
    print('The vector f is not solenoidal')
```

<u>OUT PUT</u>

```
Divergence of the vector f is
```

$$\frac{\partial}{\partial x}\left(-x^2 + yz\right) + \frac{\partial}{\partial y}\left(4y - z^2\right) + \frac{\partial}{\partial z}(2xz - 4z)$$

$$0$$

```
The vector f is solenoidal
```

---

7 a Python program to verify whether the vector $\vec{F} = x^2yz\hat{i} + xy^2z\hat{j} + xyz^2\hat{k}$ is irrotational.

```python
from sympy . physics . vector import *
from sympy import *
x,y,z=symbols('x,y,z')
v= ReferenceFrame ('v')
f1=x**2*y*z
f2=x*y**2*z
f3=x*y*z**2
f=f1*v.x+f2*v.y+f3*v.z
print (" curl of f is ")
display((Derivative(f3,y)-Derivative(f2,z))*v.x-(Derivative(f3,x)-Derivati
curlf=(diff(f3,y)-diff(f2,z))*v.x-(diff(f3,x)-diff(f1,z))*v.y+(diff(f2,x)-(
display (curlf)
if curlf==0:
    print('The vector f is Irrotational')
else:
    print('The vector f is not Irrotational')
```

<u>OUT PUT</u>

		curl of f is $(\frac{\partial}{\partial y}xyz^2 - \frac{\partial}{\partial z}xy^2z)\hat{\mathbf{v}}_\mathbf{x} + (-\frac{\partial}{\partial x}xyz^2 + \frac{\partial}{\partial z}x^2yz)\hat{\mathbf{v}}_\mathbf{y} + (\frac{\partial}{\partial x}xy^2z - \frac{\partial}{\partial y}x^2yz)\hat{\mathbf{v}}_\mathbf{z}$ $(-xy^2 + xz^2)\hat{\mathbf{v}}_\mathbf{x} + (x^2y - yz^2)\hat{\mathbf{v}}_\mathbf{y} + (-x^2z + y^2z)\hat{\mathbf{v}}_\mathbf{z}$ The vector f is not Irrotational

8	a	Python program to evaluate the integral $\int_0^{\log 2}\int_0^{x}\int_0^{x+\log y} e^{x+y+z}\,dz dy dx$
		`from sympy import *`
		`x,y,z=symbols('x,y,z')`
		`f=exp(x+y+z)`
		`I=Integral(f,(z,0,x+log(y)),(y,0,x),(x,0,log(2)))`
		`display(I)`
		`I=integrate(f,(z,0,x+log(y)),(y,0,x),(x,0,log(2)))`
		`I=simplify(I)`
		`display(I)`
		
		<u>OUT PUT</u>
		$\int\limits_0^{\log(2)}\int\limits_0^{x}\int\limits_0^{x+\log(y)} e^{x+y+z}\,dz\,dy\,dx$
		$-\frac{19}{9} + \frac{8\log(2)}{3}$

	b	Python program to find the area bounded by the parabolas $y^2 = 4ax$ and $x^2 = 4ay$
		`from sympy import *`
		`x,y,a=symbols('x,y,a')`
		`A=Integral(1,(y,x**2/(4*a),2*sqrt(a*x)),(x,0,4*a))`
		`display(A)`
		`A1=integrate(1,(y,x**2/(4*a),2*sqrt(a*x)),(x,0,4*a))`
		`display(A1)`
		
		<u>OUT PUT</u>

$$\int_{0}^{4a}\int_{\frac{x^2}{4a}}^{2\sqrt{ax}} 1\,dy\,dx$$

$$-\frac{16a^2}{3}+\frac{32\left(a^2\right)^{\frac{3}{2}}}{3a}$$

9 a

Python program to evaluate the integral $\int_0^1\int_0^{\sqrt{1-y^2}} x^3 ydxdy$

```
from sympy import *
x,y=symbols('x,y')
f=(x**3)*y
I=Integral(f,(x,0,sqrt(1-y**2)),(y,0,1))
display(I)
I=integrate(f,(x,0,sqrt(1-y**2)),(y,0,1))
print('I=',I)
```

<u>OUT PUT</u>

$$\int_{0}^{1}\int_{0}^{\sqrt{1-y^2}} x^3 y\,dx\,dy$$

```
I= 1/24
```

b

Python program to find the volume of the solid bounded by the planes $x=0, y=0,\ \ z=0$ and $\frac{x}{a}+\frac{y}{b}+\frac{z}{c}=1$.

```
from sympy import *
x,y,z,a,b,c=symbols('x,y,z,a,b,c')
A=Integral(1,(z,0,c*(1-(x/a)-(y/b))),(y,0,b*(1-(x/a))),(x,0,a))
display(A)
A1=integrate(1,(z,0,c*(1-(x/a)-(y/b))),(y,0,b*(1-(x/a))),(x,0,a))
display(A1)
```

<u>OUT PUT</u>

		$\int_0^a \int_0^{b(1-\frac{x}{a})} \int_0^{c(1-\frac{y}{b}-\frac{x}{a})} 1\, dz\, dy\, dx$ $\frac{abc}{6}$

10	a	Python program to find the area bounded by the curves $y = x^2$ and $y = x$
		
		`from sympy import *`
		`x,y=symbols('x,y')`
		`A=Integral(1,(y,x**2,x),(x,0,1))`
		`display(A)`
		`A1=integrate(1,(y,x**2,x),(x,0,1))`
		`display(A1)`
		
		<u>OUT PUT</u>
		$\int_0^1 \int_{x^2}^{x} 1\, dy\, dx$
		
		$\frac{1}{6}$

	b	Python program to verify $\beta(m,n) = \frac{\Gamma(m)\Gamma(n)}{\Gamma(m+n)}$ for $m = \frac{3}{2}$ and $n = \frac{1}{2}$.
		
		`from sympy import *`
		`n=float(input('Enter the value of n to find Γ(n):'))`
		`m=float(input('Enter the value of m to find β(m,n):'))`
		`gn=gamma(n)`
		`gm=gamma(m)`
		`gmn=gamma(m+n)`
		`print(f'Γ({m})=%0.4f'%gm,f'\nΓ({n})=%0.4f'%gn,f'\nΓ({m}+{n})=%0.4f'%gmn)`
		`bmn=round(beta(m,n),4)`
		`print(f'β({m},{n})=%0.4f'%bmn)`
		`rhs=round((gn*gm)/gmn,4)`
		`print('R.H.S = %0.4f'%rhs)`
		`if bmn==rhs:`
		`    print('β(m,n)=(Γ(m)*Γ(n))/Γ(m+n)')`
		`else:`
		`    print('β(m,n)!=(Γ(m)*Γ(n))/Γ(m+n)')`
		
		<u>OUT PUT</u>

```
Enter the value of n to find Γ(n):0.5
Enter the value of m to find β(m,n):1.5
Γ(1.5)=0.8862
Γ(0.5)=1.7725
Γ(1.5+0.5)=1.0000
β(1.5,0.5)=1.5708
R.H.S = 1.5708
β(m,n)=(Γ(m)*Γ(n))/Γ(m+n)
```