Using the Mathematical tools, write the codes to find the gradient of $\phi = xy^2z^3$

~~from sympy.vector import from sympy~~

```
from sympy.vector import *
from sympy import symbols
N= CoordSys3D('N')
x,y,z = symbols('x y z')
A = N.x*N.y**2*N.z**3
delop = Del()
display(delop(A))
gradA = gradient(A)
```

Using Mathematical tools, write the code to find curl of

$\vec{F} = xy^2\hat{i} + 2x^2yz\hat{j} - 3yz^2\hat{k}$

(and)

$\vec{F} = x^2yz\hat{i} + yz^2x\hat{j} + y^2zx\hat{k}$

```
from sympy.vector import *
from sympy import Symbols
N= CoordSys3D('N')
x,y,z = Symbols('xyz')
A = N.x**2*N.y*N.z*N.i + N.y*N.z**2*N.x*N.j
                    +N.y**2*N.z*N.x*N.k

A= N.x*N.y**2*N.i + N.2*x**2*N.y*N.z*N.j - N.3*y
                    *N.z**2*N.k

delop = Del()
curlA = delop.cross(A)
display(curl(A))
print("\n curl of(A) is \n")
display(curl(A))
```

P
P
P
P

Using mathematical tools, write the code to find the divergence of $\vec{F} = x^2 y i + y z^2 j + x^2 z k$

```
from sympy.vector import *
from sympy import symbols
N= CoordSys3D('N')
x,y,z = symbols('x y z')
A = N.x**2*N.y*N.i + N.y*N.z**2*N.j + N.x**2
    *N.z*N.k

delop = Del()
divA = delop.dot(A)
divergA = divergence(A)
display(divA)
print("\n Divergence of (A) is \n")
display(divergA)
```

Using Mathematical tools, write the code to find the sol^n of ~~Repeated~~ $\frac{dy}{dx} = 1 + \frac{y}{x}$ $y(1)=2$ at $y(2)$ taking $h = 0.2$ by RKM 4th order

$\frac{dy}{dx} = x - y^2$ at $y(0.2)$, $y(0) = 1$, $h = 0.2$

```
from sympy import *
import numpy as np
def Rungekutta (g, x0, h, y0, xn):
    x, y = symbols('x y')
    f = lambdify ([x, y], g)
    xt = x0+h
    Y=[y0]
    while xt<=xn
        k1 = h*f(x0, y0)
        k2 = h*f(x0+h/2, y0+k1/2)
        k3 = h*f(x0+h/2, y0+k2/2)
        k4 = h*f(x0+h, y0+k3)
        y1 = y0+(1/6)*(k1+2*k2+2*k3+k4)
        Y.append(y1)
        x0 = xt
        y0 = y1
        xt = xt+h
    return np.round(Y, 2)
Rungekutta ('1+(y/x)', 1, 0.2, 2, 2) ['x-(y**2)', 0, 0.2, 1, 0.2)
```

Using Mathematical tool, write the code solve the differential equation $\frac{dy}{dx} = 3e^x + 2y$ with $y(0) = 0$, Using the Taylor's series method at $x = 0.1(0.1)0.3$

```
from numpy import array
def taylor (deriv, x, y, xStop, h):
        x = [ ]
        y = [ ]
        x.append(x)
        y.append(x)
        while x < xStop
                D = deriv(x, y)
        H = 1.0
        for j in range(3)
            H = H*h /(j+1)
            y = y + D[j]* H
        x = x+h
        x.append(x)
        y.append(y)
        return array(x), array(y)
def deriv(x, y):
    D = zeros((4, 1))
    D[0] = [2*y[0] + 3*exp(x)]
    D[1] = [4*y[0] + 9* exp(x)]
    D[2] = [8* y[0] + 21* exp(x)]
    D[3] = [16* y[0] + 45* exp(x)]
    return D
x = 0.0
xStop = 0.3
y = array([0.0])
h = 0.1
X, Y = taylor (deriv, x, y, xStop, h)
print("The required values are :at x=%0.2f, y=%0.5f, x=%0.2f
y=%0.5f, x=%0.2f, y=%0.5f, x=%0.2f, y=%0.5f
" %(x[0], y[0], x[1], y[1], x[2], y[2], x[3], y[3]))
```