Question 1

```
# part (i):

# defining a function to return f(x) values
def f1(x):
  return ((x**3) - (3*(x**2)) + x - 1)

# now defining a function to return the g(x) values from x = g(x) form of f(x)
def g1(x):
  return (3 - 1/(x) + 1/(x**2))

# now defining the fixed point iteration function
def fixed_point1(x0):
    count = 1
    flag = 1
    while True:
        x1 = g1(x0)
        print('x = %0.6f and f(x) = %0.6f' % (x1, f1(x1)))
        count += 1
        if abs(x1 - x0) < 0.000001:
            break
        elif count > 10000:
            flag=0
            break

        x0 = x1

    if flag==1:
        print('\nroot is: %0.6f' % x1)
    else:
        print('\Does not converge.')

x0 = 2    # choosing 2 as the initial value as |g'(x)| at 2 is less than 1
fixed_point1(x0)
# Output is below:
```

```
x = 2.750000 and f(x) = -0.140625
x = 2.768595 and f(x) = -0.005151
x = 2.769267 and f(x) = -0.000187
x = 2.769291 and f(x) = -0.000007
x = 2.769292 and f(x) = -0.000000

root is: 2.769292
```

```
# part (ii) Q1:

from math import sin, exp, log, pi, asin

# defining a function to return f(x) values
def f2(x):
  return (sin(x) - exp(-x))

# now defining g(x) as sin inverse of e^-x:
def g2(x):
  return asin(exp(-x))

# now defining the fixed point iteration function
def fixed_point2(x0):
    count = 1
    flag = 1
    while True:
        x1 = g2(x0)
        print('x = %0.6f and f(x) = %0.6f' % (x1, f2(x1)))
        count += 1
        if abs(x1 - x0) < 0.000001:
            break
        elif count > 10000:
            flag=0
            break

        x0 = x1

    if flag==1:
        print('\nroot is: %0.6f' % x1)
    else:
        print('\Does not converge.')

x0 = 0.5     # choosing 0.5 as the initial value as |g'(x)| at 0.5 is less than 1
fixed_point2(x0)
#Output is below:

     x = 0.651690 and f(x) = 0.085366
     x = 0.548215 and f(x) = -0.056816
     x = 0.616252 and f(x) = 0.038016
     x = 0.570395 and f(x) = -0.025338
     x = 0.600800 and f(x) = 0.016929
     x = 0.580417 and f(x) = -0.011292
     x = 0.593981 and f(x) = 0.007540
     x = 0.584911 and f(x) = -0.005031
```

```
x = 0.590957 and f(x) = 0.003358
x = 0.586918 and f(x) = -0.002241
x = 0.589612 and f(x) = 0.001496
x = 0.587813 and f(x) = -0.000998
x = 0.589013 and f(x) = 0.000666
x = 0.588212 and f(x) = -0.000445
x = 0.588747 and f(x) = 0.000297
x = 0.588390 and f(x) = -0.000198
x = 0.588628 and f(x) = 0.000132
x = 0.588469 and f(x) = -0.000088
x = 0.588575 and f(x) = 0.000059
x = 0.588504 and f(x) = -0.000039
x = 0.588552 and f(x) = 0.000026
x = 0.588520 and f(x) = -0.000018
x = 0.588541 and f(x) = 0.000012
x = 0.588527 and f(x) = -0.000008
x = 0.588536 and f(x) = 0.000005
x = 0.588530 and f(x) = -0.000003
x = 0.588534 and f(x) = 0.000002
x = 0.588532 and f(x) = -0.000002
x = 0.588533 and f(x) = 0.000001
x = 0.588532 and f(x) = -0.000001
x = 0.588533 and f(x) = 0.000000

root is: 0.588533
```

Question 2

```
from math import cos

# defining the function to return f(x) values
def f3(x):
  return (cos(x) - 3*x + 1)

# defining the derivative of f(x)
def f_derv(x):
  return (-sin(x) - 3)

# now defining the Newton-Raphson function:
def newton_raphson(x0):
    x = x0
    while True:
        h = -f3(x)/f_derv(x)
        x += h
        print('x = %0.6f and f(x) = %0.6f' % (x, f3(x)))
        if (abs(h) <= 0.0000001):
          break
```

```
    print("Root is : ", "%.6f"% x)

x0 = 0.5  # choosing 0.5 as the initial guess
newton_raphson(x0)
# Output is below:
```

```
x = 0.608519 and f(x) = -0.005060
x = 0.607102 and f(x) = -0.000001
x = 0.607102 and f(x) = -0.000000
x = 0.607102 and f(x) = 0.000000
Root is :  0.607102
```

✓ 0s completed at 4:35 PM