Assignment 1

Omar Ahmed

UCID:30154382

```python
# Question 1
k = -3
x = 0

for k in range(-3,5):
    expression = (k**2) * np.sin(0.1 * (k**2))
    k += 1
    x += expression

print(x)
```

```
PS C:\Users\omar5> & C:/Users/omar5/anaconda3/python.exe "c:/Users/omar5/Desktop/ENEL 102/1/test.py"
33.40807559372164
```

```python
# Question 2
j = 1
k = -3
x = 0

for j in range(1,4):
    for k in range(-3,5):
        expression = np.sqrt(j) * (k**2) * np.sin(0.1*((k-j)**2))
        k += 1
        x += expression
    j += 1

print(x)
```

```
PS C:\Users\omar5> & C:/Users/omar5/anaconda3/python.exe "c:/Users/omar5/Desktop/ENEL 102/1/test.py"
56.671408415734454
```

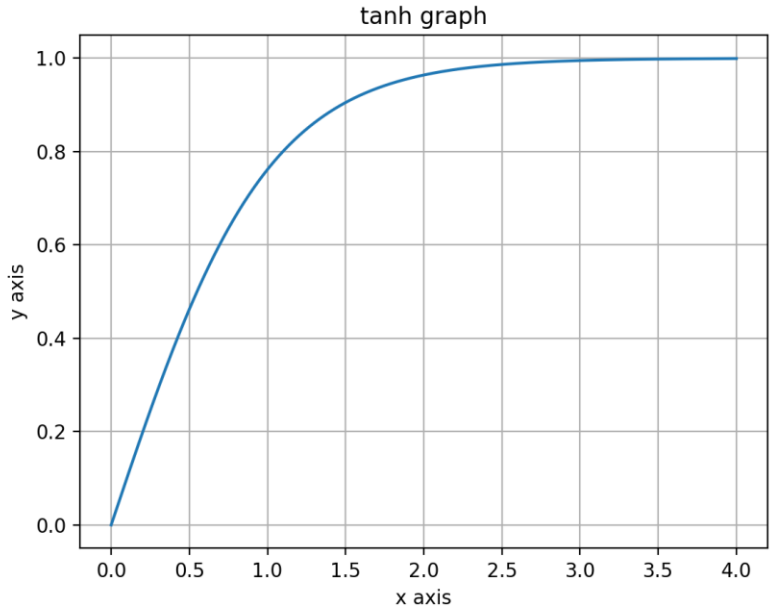
```python
# Question 3
x = np.sqrt(3)
y = (0.3 * x**2) + np.sqrt(x)
z = np.sqrt(np.exp(1)) + x - np.log(x)- np.log10(x)
v = np.sqrt(np.tanh(x*y*z))

print(v)
```

```
0.9999999977333676
```

```
# Question 4
x = np.linspace(0,4,500)
y = np.tanh(x)

plt.plot(x,y)
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("tanh graph")
plt.grid()
plt.show()
```




```
# Question 5a
x = -4 + 1j
y = 3j
z = np.array([x**y, x*(y**2), np.exp(np.sqrt(x))])
m2 = np.linalg.norm(np.array([x**y, x*(y**2), np.exp(np.sqrt(x))])) ** 2

print(m2)
```

```
1378.6424627693334
```

```
# Question 5b
x = -4 + 1j
y = 3j
z = np.array([x**y, x*(y**2), np.exp(np.sqrt(x))])
m = np.abs(z)
p = np.angle(z)

print('m = ', m)
print('p = ', p)
```

```
m =  [1.68286618e-04 3.71079506e+01 1.28158603e+00]
p =  [-2.03336529 -0.24497866  2.01532946]
```

```
# Question 6
x = np.array([[1, 2, -3], [4, 8, 8], [2, 2, 4]])
y = x + np.matmul(np.transpose(x), x) + np.linalg.matrix_power(x,3)
print(y)
```

```
[[  75  144  125]
 [ 614 1056  958]
 [ 221  380  367]]
```

```
# Question 7
a = np.array([[1, 2, -3], [4, 8, 8], [2, 2, 4]])
b = np.array([[5, 5, -3], [4, 8, 8], [2, 2, 4]])
z = np.zeros((3,3), dtype = int)

aa = np.concatenate((np.concatenate((a, b), axis = 1), np.concatenate((z, a), axis = 1)))
bb = np.array([[1], [0], [0], [0], [0], [0]])

x = np.linalg.solve(aa,bb)

print(x)
```

```
[[ 0.4]
 [-0. ]
 [-0.2]
 [ 0. ]
 [-0. ]
 [-0. ]]
```

```python
# Question 8
x = np.arange(-50, 31)
y = 3 * np.power(x, 2) + 2
q = np.array([x, y])
qt = np.transpose(q)

z = np.dot(q , qt)
print(z)
```

```
[[    52380  -4229820]
 [ -4229820 639094860]]
```

```python
# Question 9
u = np.array([-3, 4, -2])
v = np.array([2, -5, -4])
w = np.array([1, -1, -1])

q = np.power(np.dot(u,v), 2) + np.linalg.norm((np.cross(np.cross(u,v),w)))

print(q)
```

```
375.51698748956505
```

```python
# Question 10
x = np.array([[1, 2, 3], [0, 7, 7], [1, 2, 1]])
y = np.array([[2, 2, 3], [7, 6, 0], [1, 2, 1]])

q = np.matmul(np.linalg.inv(x), (y + np.linalg.matrix_power(x,2)))

print(q)
```

```
[[0.5        2.28571429 5.        ]
 [0.5        7.85714286 6.        ]
 [1.5        2.         2.        ]]
```

```python
# Question 11
# system of equations
# 4x + y + z = 3
# 2x + 2 + 13z = 4 - y (2x + y + 13z = 2)
# 3x - z + 3y = 11 + 3y (3x - z = 11)

a = np.array([[4, 1, 1], [2, 1, 13], [3, 0, -1]])
b = np.array([3, 2, 11])
q = np.linalg.solve(a,b)

print(np.reshape(q,(3,1)))
```

```
[[  3.85294118]
 [-12.97058824]
 [  0.55882353]]
```

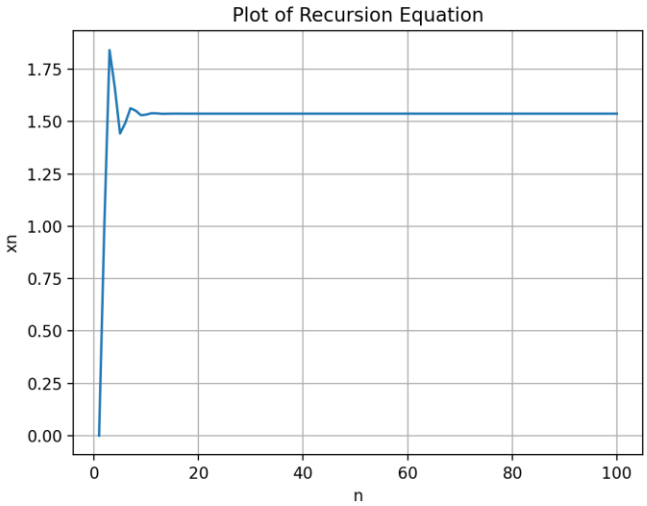
```python
# Question 12
# x1 = x0 = 0
# for 101 points

n = np.linspace(1,100,100)
x = np.zeros(101)

for i in range(2,101):
    x[i] = np.sin(x[i-1]) - 0.3 * x[i-2] + 1

# plot for 1-100
yaxis = np.delete(x,0)
xaxis = n
print(x)

plt.plot(xaxis, yaxis)
plt.xlabel('n')
plt.ylabel('xn')
plt.title('Plot of Recursion Equation')
plt.grid()
plt.show()
```


Plot of Recursion Equation
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
xn
0
20
40
60
80
100
n