QuestionOne

```
import math
x = sum([(k**2)*math.sin(0.1*(k**2)) for k in range(-3,5)])
print(f"{x = }")
```

```
x = 33.40807559372164
```

QuestionTwo

```
import math
x = sum([sum([math.sqrt(j)*(k**2)*math.sin(0.1*((k-j)**2)) for k in
range(-3, 5)]) for j in range(1, 4)])
print(f"{x = }")
```

```
x = 56.67140841573445
```

QuestionThree

```
import math
x = math.sqrt(3)
y = 0.3*(x**2) + math.sqrt(x)
z = math.sqrt(math.e) + x - math.log(x) - math.log(x, 10)
v = math.sqrt(math.tanh(x*y*z))
print(f"{v = }")
```

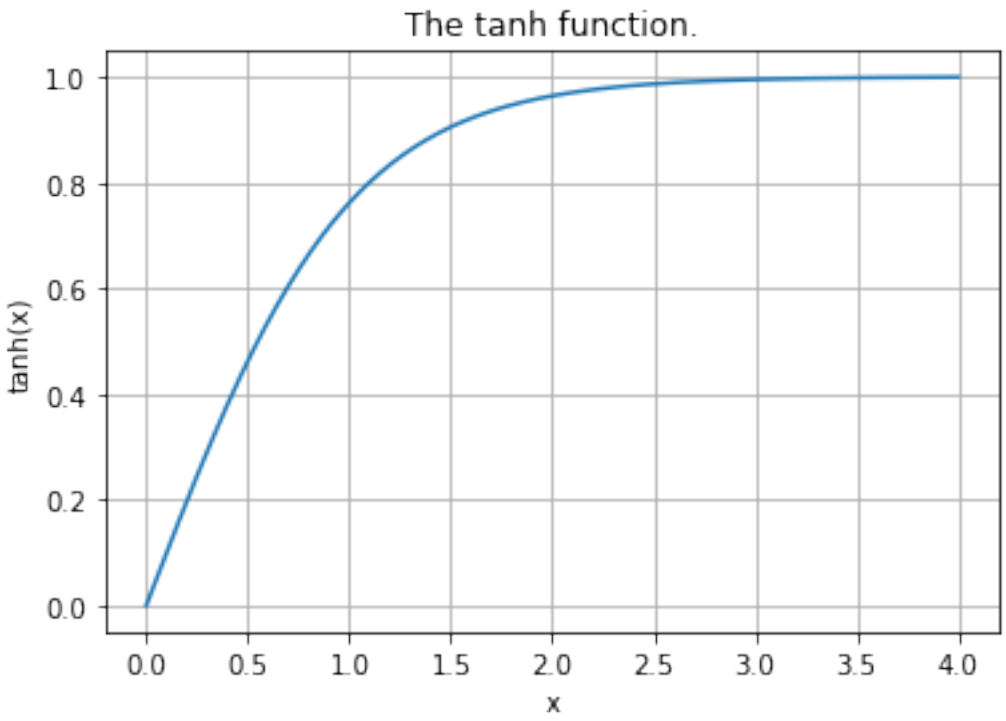
```
v = 0.9999999977333676
```

QuestionFour

```
import matplotlib.pyplot as plt
import numpy as np

# There was a typo in the question should be from 0 to 4 with 500
points.
x = np.linspace(0, 4, 500)
y = np.tanh(x)

plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('tanh(x)')
plt.title('The tanh function.')
plt.grid()
plt.show()
```



QuestionFive

```
import numpy as np

x = -4 + 1j
y = 3j
z = [pow(x,y), x*(pow(y,2)), pow(np.e, (np.sqrt(x)))]

print("Part A:")
mag_square = sum([(np.abs(i)**2) for i in z])
print(f"magnitude squared of z = {mag_square}")
print("Part B:")

m = np.abs(z)
print(f"{m = }")
p = np.angle(z)
print(f"{p = }")
```

```
Part A:
magnitude squared of z = 1378.6424627693336
Part B:
m = array([1.68286618e-04, 3.71079506e+01, 1.28158603e+00])
p = array([-2.03336529, -0.24497866,  2.01532946])
```

QuestionSix

```
import numpy as np
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(f"{y = }")

y = array([[  75,  144,  125],
       [ 614, 1056,  958],
       [ 221,  380,  367]])
```

QuestionSeven

```
import numpy as np
A = np.array([[1,2,-3], [4,8,8], [2,2,4]])
B = np.array([[5,5,-3], [4,8,8], [2,2,4]])
zeros = np.zeros((3,3)).astype(np.int32)
x = np.linalg.solve(
    np.vstack((np.hstack((A, B)), np.hstack((zeros, A)))),
    np.array([1,0,0,0,0,0]))
print(f"{x = }")

x = array([ 0.4, -0. , -0.2,  0. , -0. , -0. ])
```

QuestionEight

```
import numpy as np
x = np.linspace(-50, 30, 81).astype(np.int32)
y = 3*np.square(x) + 2
Q = np.array([x, y])
z = np.matmul(Q, Q.transpose())
print(f"{z = }")

z = array([[    52380,  -4229820],
       [ -4229820, 639094860]])
```

QuestionNine

```
import numpy as np
u = np.array([-3, 4, -2])
v = np.array([2, -5, -4])
w = np.array([1, -1, -1])
Q = np.square(np.dot(u,v)) + np.linalg.norm(np.cross(np.cross(u, v),
w))
print(f"{Q = }")

Q = 375.51698748956505
```

QuestionTen

```
import numpy as np
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.square(X))
print(f"{Q = }")
```

```
Q = array([[ 0.5       , -9.71428571, -7.        ],
       [ 0.5       ,  7.85714286,  2.        ],
       [ 0.5       ,  0.        ,  5.        ]])
```

QuestionEleven

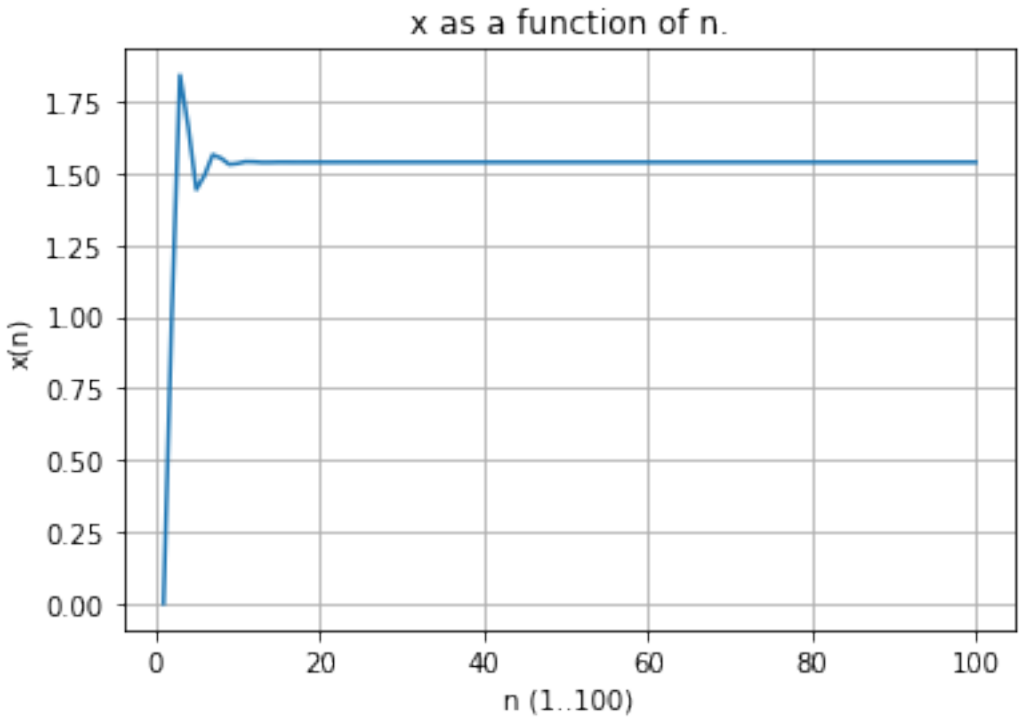
```
"""
4x + y + z = 3
2x + y + 13z = 2
3x - z = 11
================
[4, 1,  1][x]   [3]
[2, 1, 13][y] = [2]
[3, 0, -1][z]   [11]
"""
import numpy as np
Q = np.linalg.solve(np.array([[4,1,1], [2,1,13], [3,0,-1]]),
np.array([3,2,11]))
print(f"{Q = }")
```

```
Q = array([  3.85294118, -12.97058824,   0.55882353])
```

QuestionTwelve

```
import matplotlib.pyplot as plt
import numpy as np
n = np.linspace(0,100,101).astype(np.int16)
x = np.zeros(101)
for i in n:
    if i in [0, 1]:
        x[i] = 0
    else:
        x[i] = np.sin(x[i-1]) - 0.3*x[i-2] + 1

plt.plot(n[1:], x[1:])
plt.xlabel('n (1..100)')
plt.ylabel('x(n)')
plt.title('x as a function of n.')
plt.grid()
plt.show()
```


x as a function of n.
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
x(n)
0
20
40
60
80
100
n (1..100)