Department of Electrical
& Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	**Signals and Systems II**
Course Number:	**ELE632**
Semester/Year	**S2022**

Instructor:	**Prof.**

Assignment/Lab Number:	**1**
Assignment/Lab Title:	**1**

Submission Date:	**May 29th, 2022**
Due Date:	**May 29th, 2022**

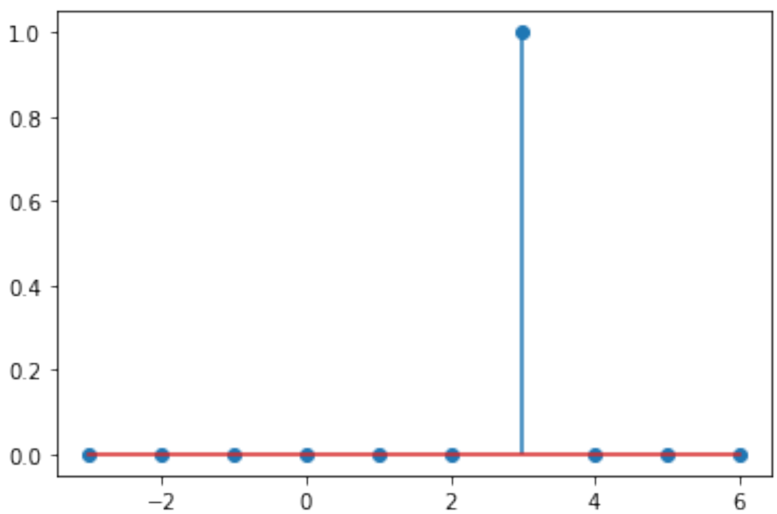
Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Aihemaiti	Ankaer	500831325	02	A.A

LAB 1: Time-Domain Analysis of Discrete-Time Systems -Part 1

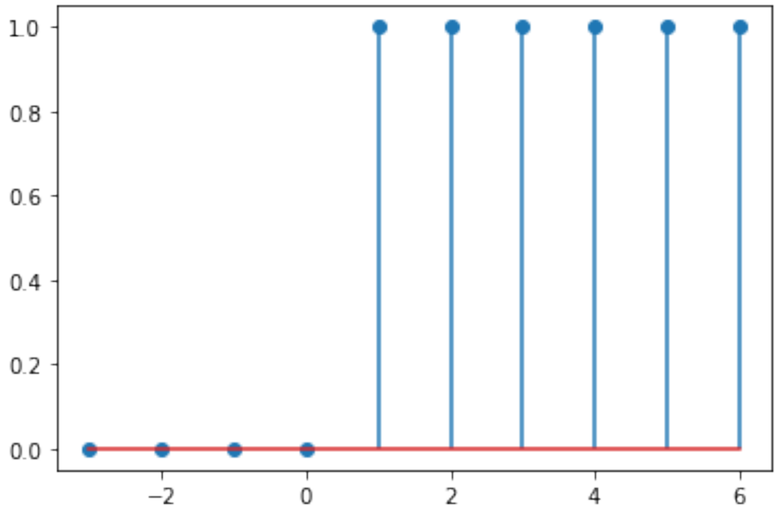
A. SignalTransformation

1)

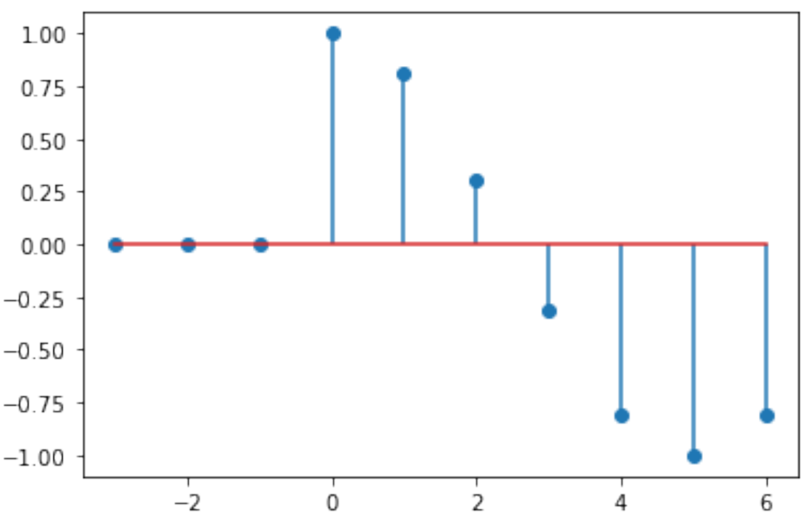
In [ ]:
```
# I. Delta[n-3]
import numpy as np
import matplotlib.pyplot as plt

def u(a, n):
    unit = np.array([])
    for sample in n:
        if sample<a:
            unit = np.append(unit,0)
        else:
            unit = np.append(unit,1)
    return(unit)



n = np. arange(-3,7)

plt.figure()
plt.stem(n, u(3,n)-u(4,n))
plt.show()
```



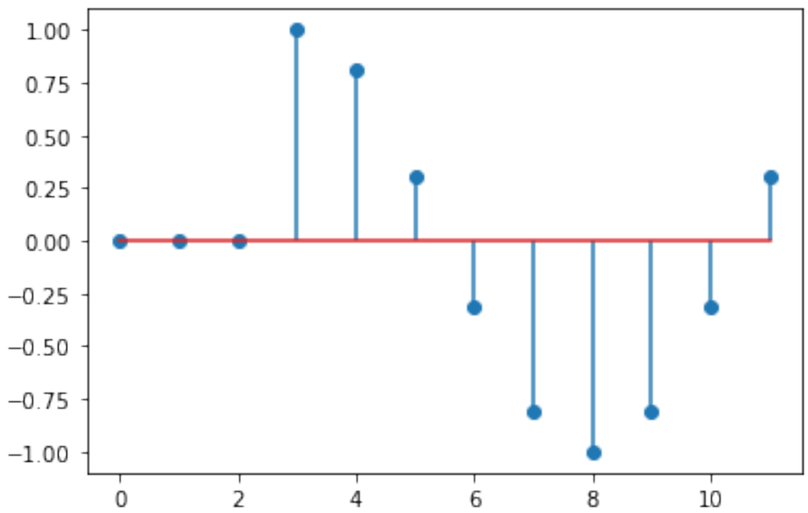
In [ ]:
```
# II. u[n+1]

plt.figure()
plt.stem(n, u(1,n))
plt.show()
```



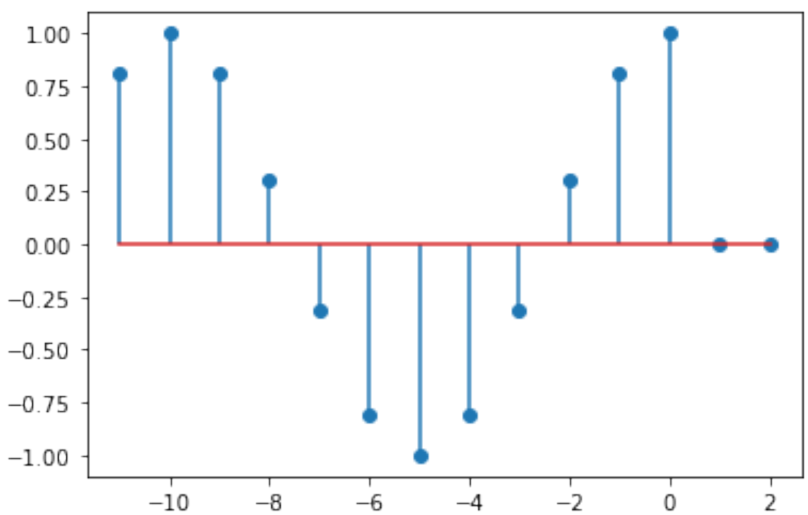
In [ ]:
```
# III. x[n] = cos(pi*n/5)u[n]

x = np.cos(np.pi*n/5)

plt.figure()
plt.stem(n,x*u(0,n))
plt.show()
```



In [ ]:
```
# IV. x1[n] = x[n-3]

n = np.arange(0,12)
x = np.cos(np.pi*(n-3)/5)

plt.figure()
plt.stem(n, x*u(3,n))
plt.show()
```



In [ ]:
```
# V. x2[n] = x[-n]

n = np.arange(-11,3)
x = np.cos(np.pi*(-n)/5)

plt.figure()
plt.stem(n, x*u(0,-n))
plt.show()
```

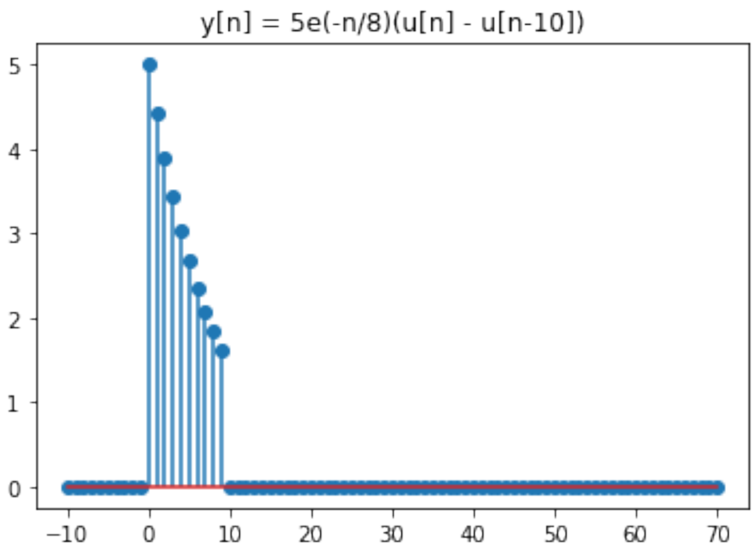


x1[n] = x[n-3] is advance by 3 x2[n] = x[-n] is reflection in the y axix

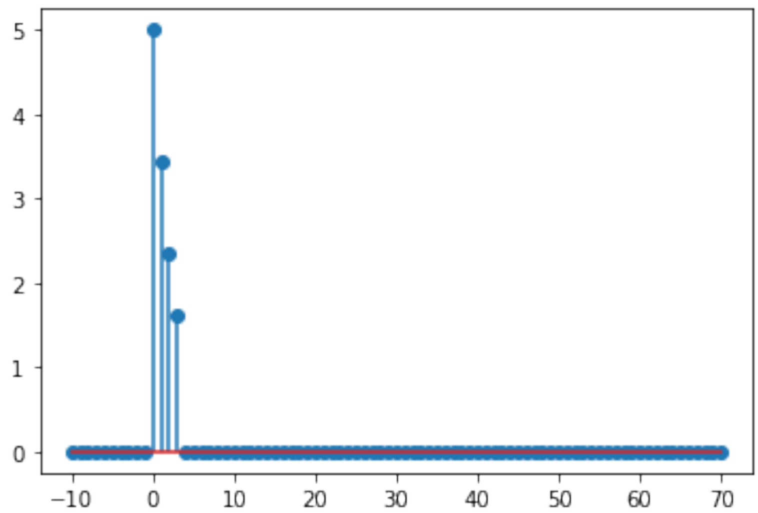
2)

In [ ]:
```
# I. y[n]=5e^(-n/8)(u[n]-u[n-10])

n = np.arange(-10,71)

y = 5*np.exp(-n*0.125)*(u(0,n)-u(10,n))

plt.figure
plt.stem(n,y)
plt.title('y[n] = 5e(-n/8)(u[n] - u[n-10])')
plt.show
```

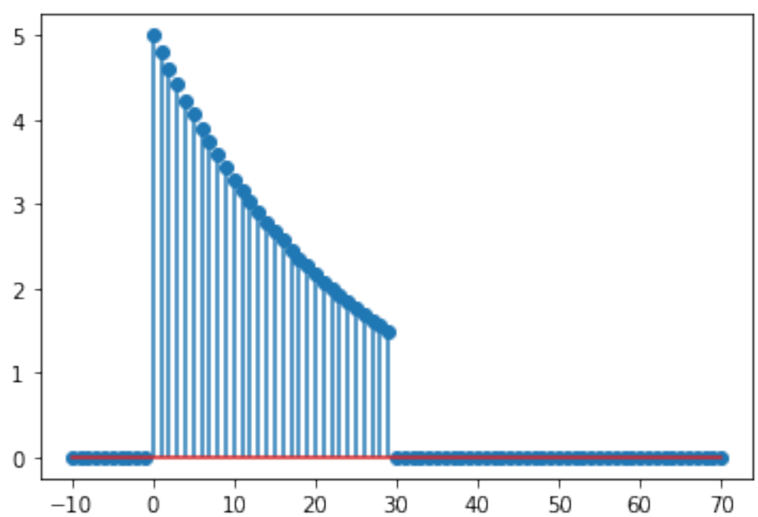
Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [ ]:
```
# II. y1[n] = y[3n]

x = 5*np.exp(-3*n/8)

plt.figure()
plt.stem(n, x*(u(0,3*n)-u(10,3*n)))
plt.show()
```



In [ ]:
```
# III. y2[n] = y[n/3]

x = 5*np.exp(-1*n/24)

plt.figure()
plt.stem(n, x*(u(0,n/3)-u(10,n/3)))
plt.show()
```

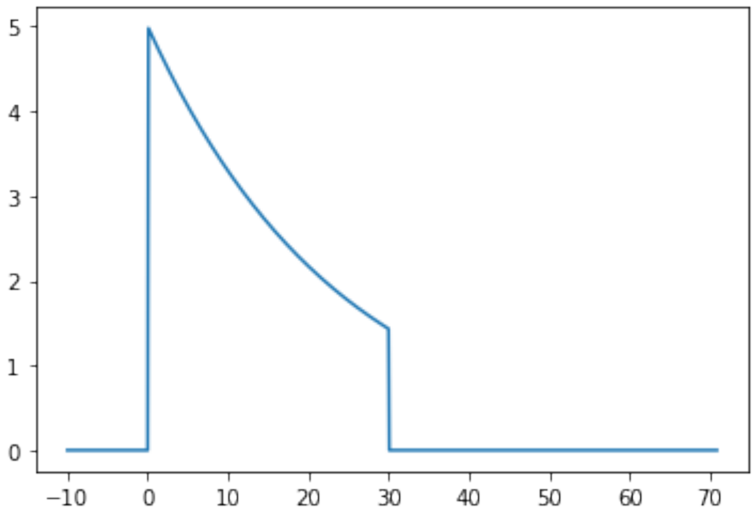


y1[n] compresses the funcitn by a factor of 3 y2[n] expands the function by a factor of 3

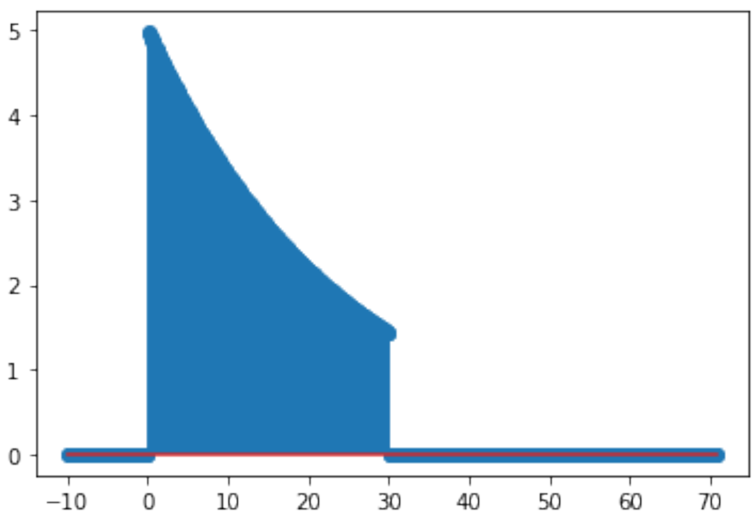
3)

In [ ]:
```
# I.
n = np.arange(-10,71,0.1)
y = 5*np.exp(-n/24)*(u(0,n/3) - u(10,n/3))

plt.figure
plt.plot(n,y)
plt.show
```

Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [ ]:
```
plt.figure()
plt.stem(n,y)
plt.show()
```



y2[n] was the linear transfomatin for the y[n] which was the sampled signa. While y3[n] was the same transformation but at the rate of 0.1 hence why it looks different.

In [ ]:
```
print('\n'*3)
```

B. RecursiveSolutionOfDifferenceEquation

1)

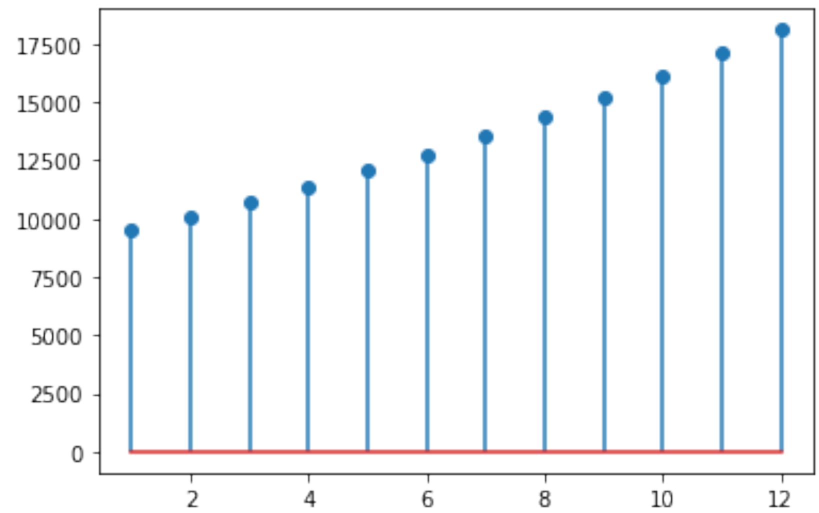
In [ ]:
```
I = 5
D = 8

n = np.arange(1,13)
r = (I + 1)*0.01
x =  200
y[0] = (D + 1)*1000

print(type(y[0]))


for i in n:
    y[n] = (1+r)*y[n-1]+x  # The equation relating the output y[n] to the ir
    output = y[n]
```

```
<class 'numpy.float64'>
```

2)

In [ ]:
```
for i in n:
    y[n] = (1+r)*y[n-1]
    output = y[n]


plt.figure
plt.stem(n,output)
plt.show
```

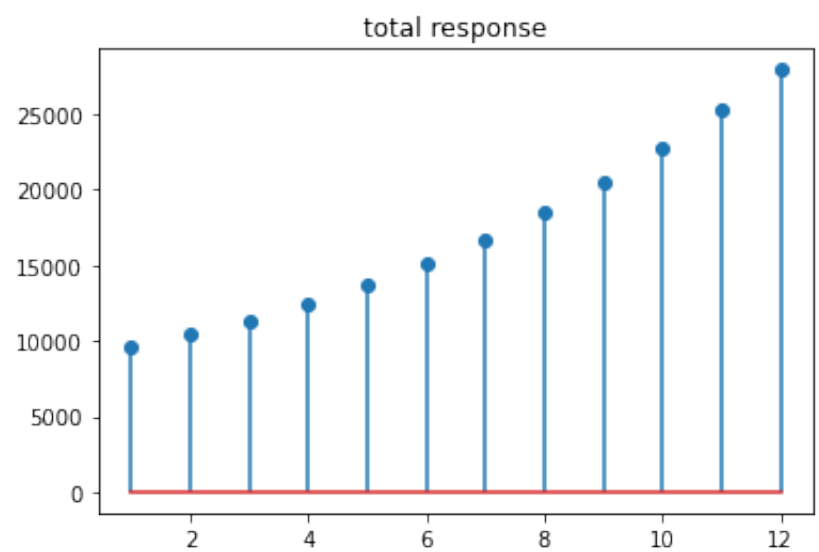
Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



3)

In [ ]:
```
for i in n:
    y[n] = (1+r)*y[n-1] + 100*n
    output = y[n]

plt.figure
plt.stem(n,output)
plt.title('total response')
plt.show
```

Out[ ]: <function matplotlib.pyplot.show(close=None, block=None)>



In [ ]:
```
print('\n'*3)
```

C. Design of a filter: N-point maximum filter

In [ ]:

In [ ]:

D. Energy and power of a discrete signal

1)

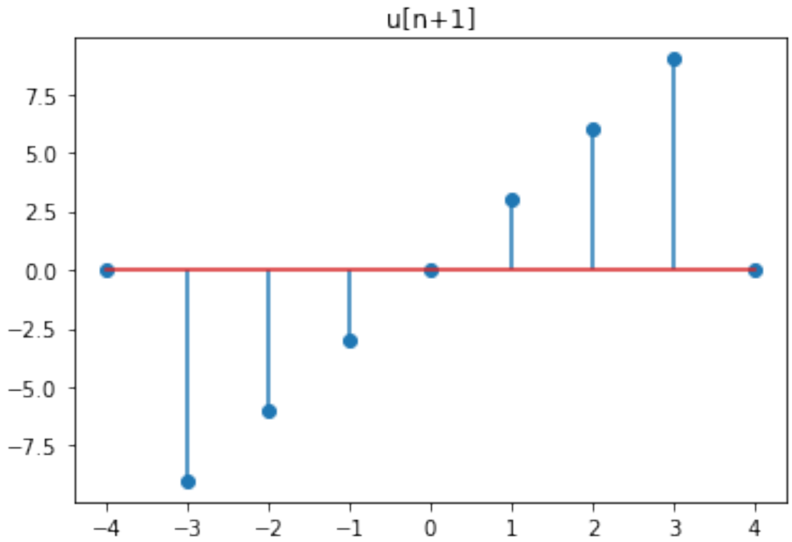
In [ ]:
```
def energy(x):
    sum = 0
    for i in x:
        term = np.absolute(x)**2
        sum += term
    return sum


def power(x, N):
    sum = 0
    for i in x:
        term = np.absolute(x)**2/(2*N+1)
        sum += term
    return sum
```

2)

In [ ]:
```
y = 3*n*(u(0,n)-u(4,n)) - 3*n*(-u(0,-n)+ u(4,-n))
n = np.arange(-4,5)
print(energy(y))
print(power(y,3))


plt.figure()
plt.stem(n,3*n*(u(0,n)-u(4,n)) - 3*n*(-u(0,-n)+ u(4,-n)))
plt.title("u[n+1]")
plt.show
```

```
[108. 432. 972.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
[ 15.42857143  61.71428571 138.85714286   0.           0.
   0.           0.           0.           0.           0.
   0.           0.        ]
```

Out[ ]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```


u[n+1]
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−4
−3
−2
−1
0
1
2
3
4