191IT234_Niraj_Nandish

August 18, 2020

1 Lab Week 1

1.0.1 Name: Niraj Nandish

1.0.2 Roll no.: 191IT234

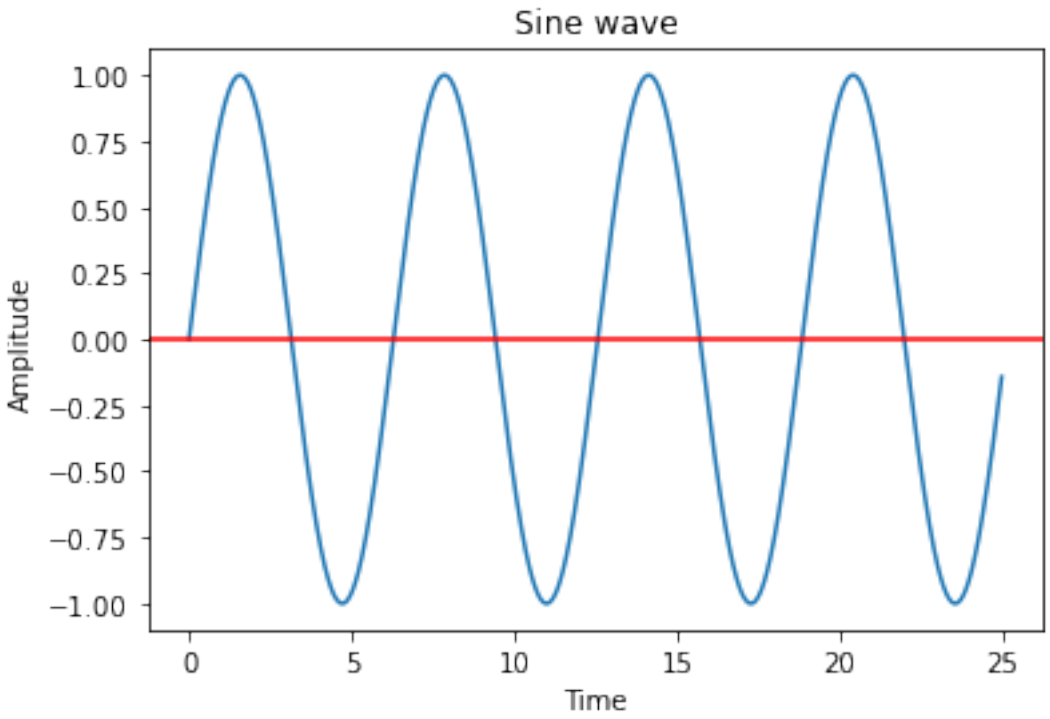
1.0.3 Semester: 3

1.1 Question 1

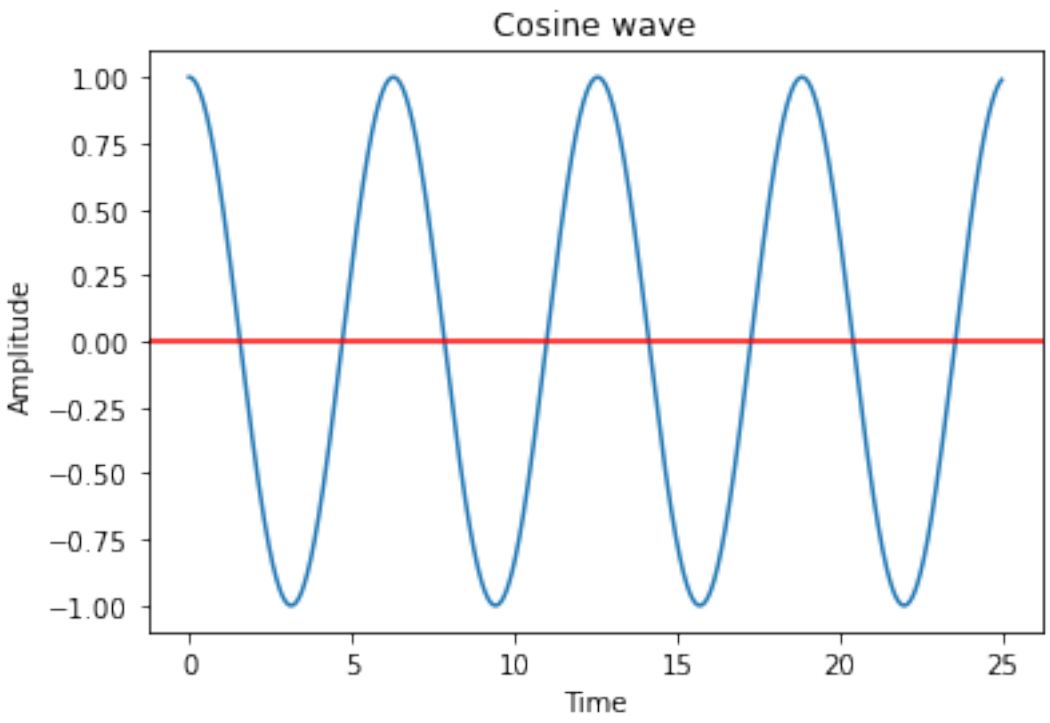
Generation of continuous signals using Python(Note: use plot function)

[1]:
```
# Import all required libraries
import numpy as np                # Contains built-in functions to work on arrays
import matplotlib.pyplot as plt   # Use to plot graphs
from scipy import signal          # Generate signals of various types
import math                       # Contains built-in math functions
```

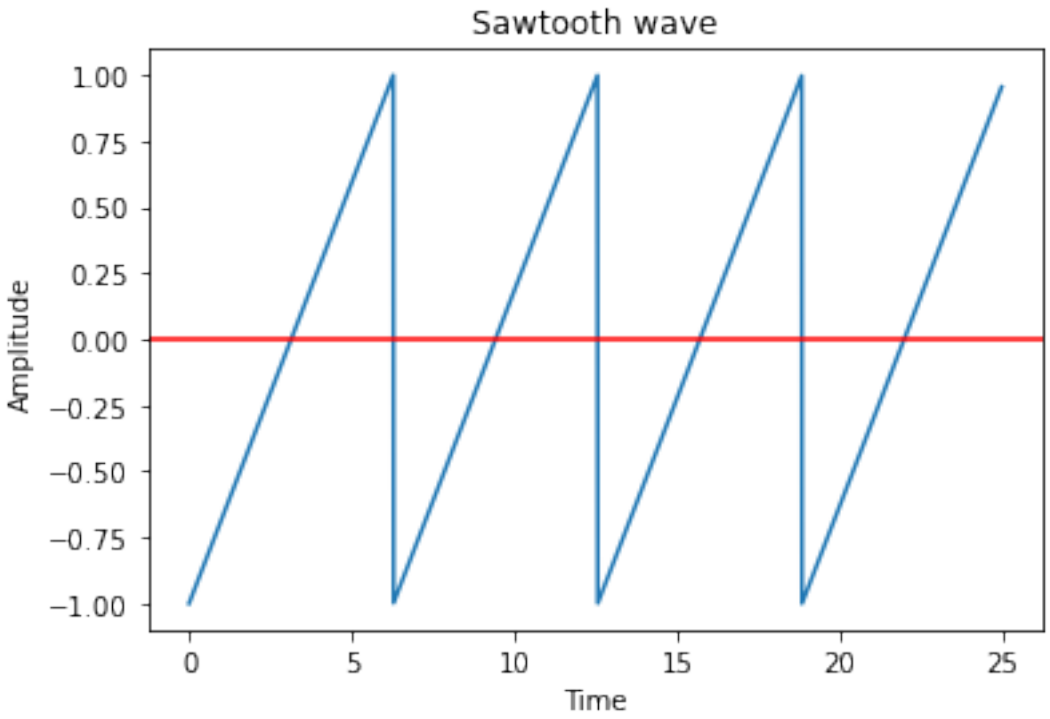
[2]:
```
# 1. sine wave

time = np.arange(0,25,0.01)
amp = np.sin(time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Sine wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



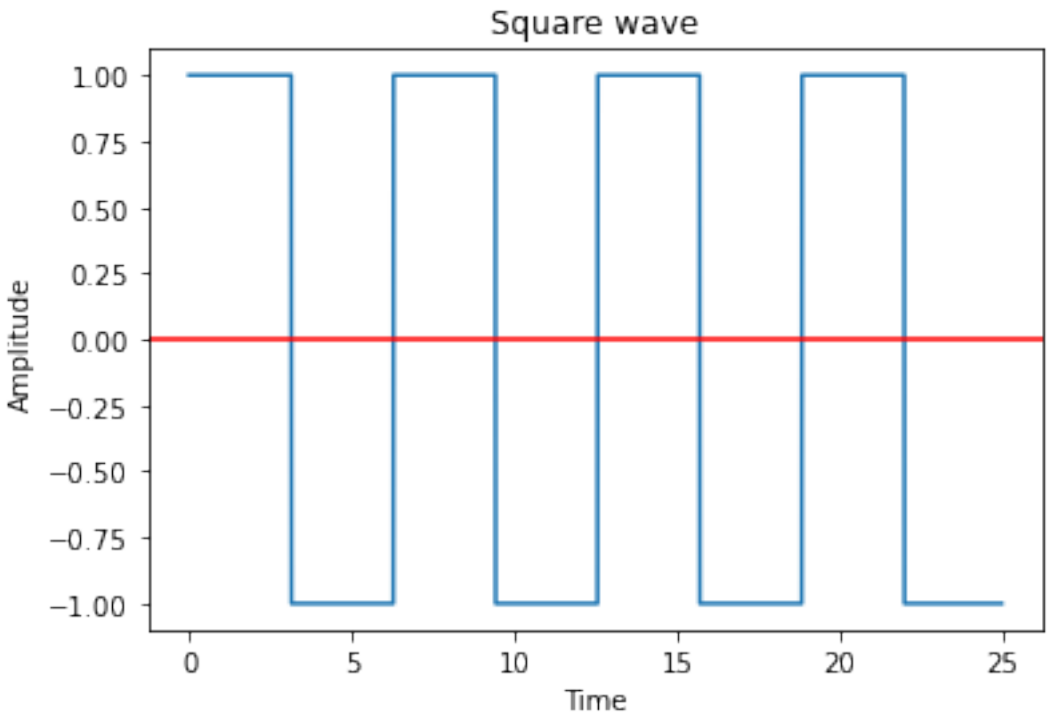
[3]:
```
# 2. cosine wave

time = np.arange(0,25,0.01)
amp = np.cos(time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Cosine wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



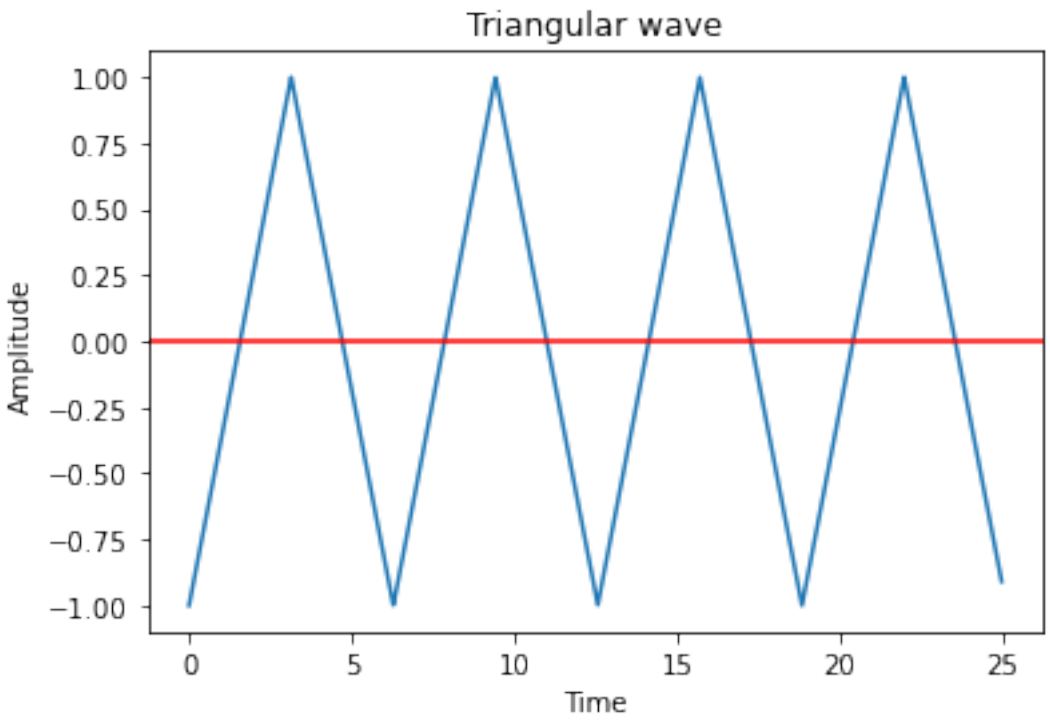
[4]:
```
# 3. sawtooth wave

time = np.arange(0,25,0.01)
amp = signal.sawtooth(time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Sawtooth wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



[5]:
```
# 4. square wave

time = np.arange(0,25,0.01)
amp = signal.square(time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Square wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



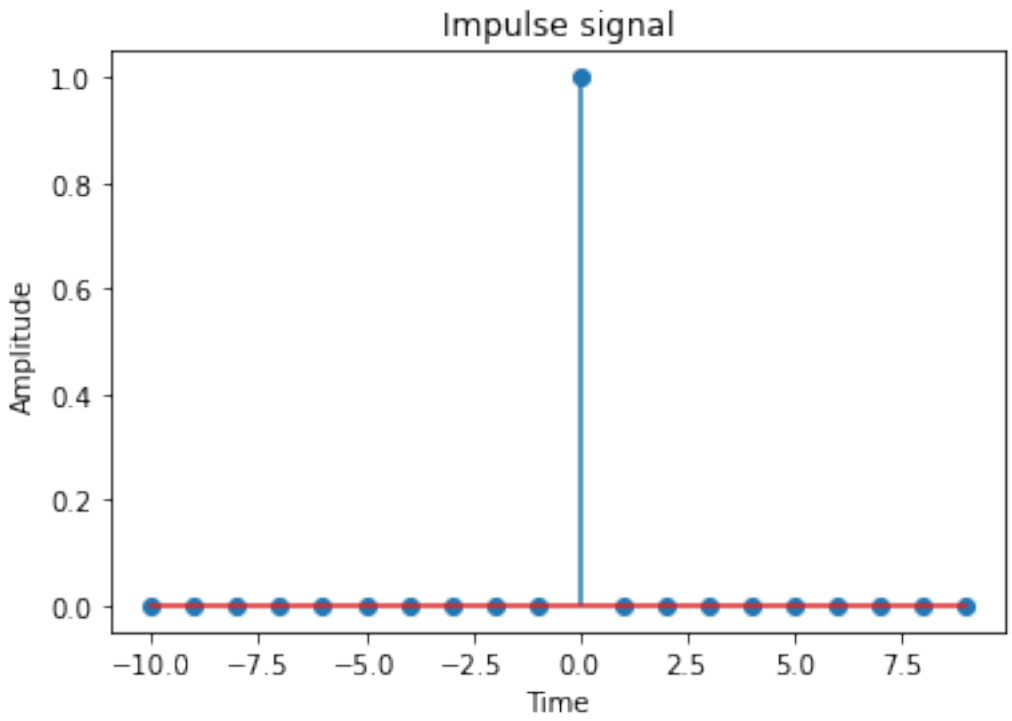
[6]:
```
# 5. triangular wave

time = np.arange(0,25,0.01)
amp = signal.sawtooth(time,0.5)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Triangular wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



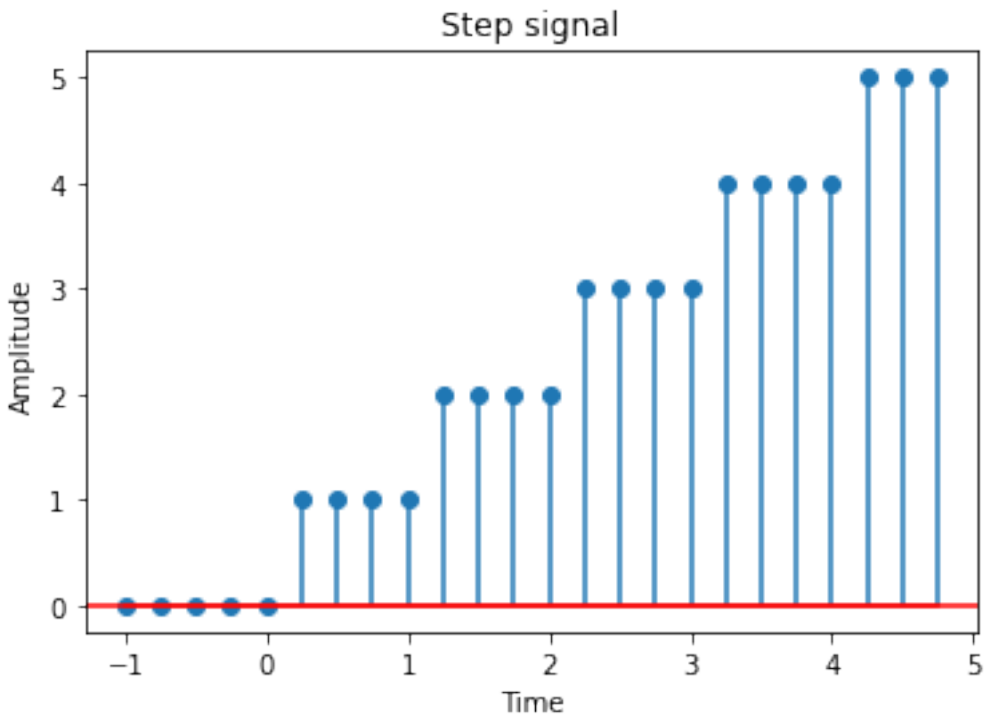
1.2 Question 2

Generation of discrete signals using Python(Note: use stem function)

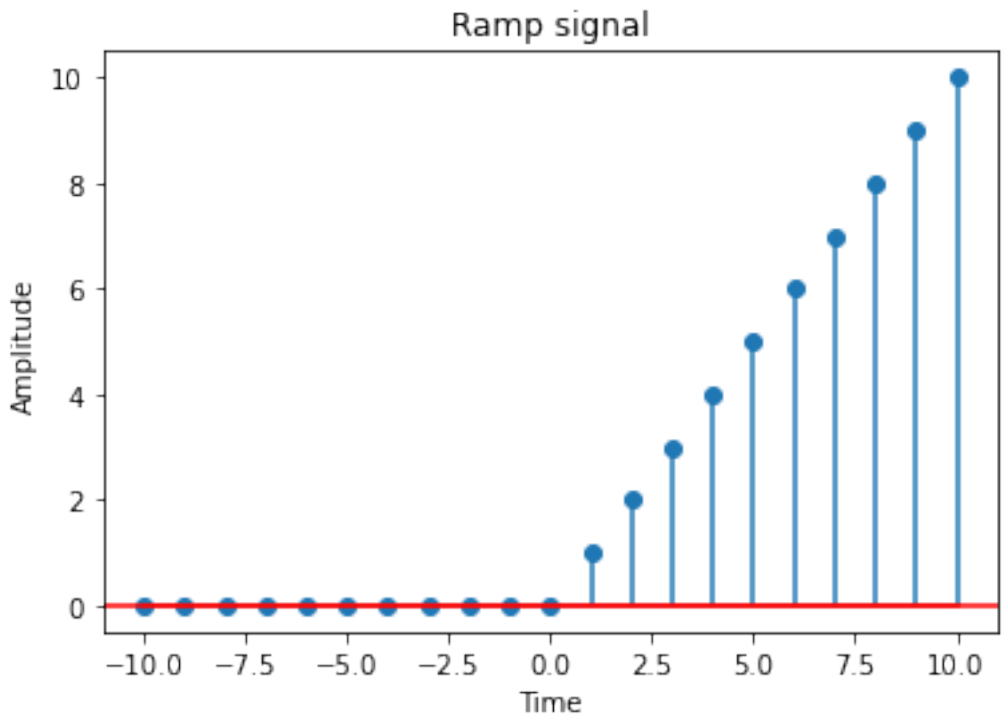
[7]:
```
# 1. Impulse signal

time = np.arange(-10,10)
amp = signal.unit_impulse(20,10)

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.title('Impulse signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



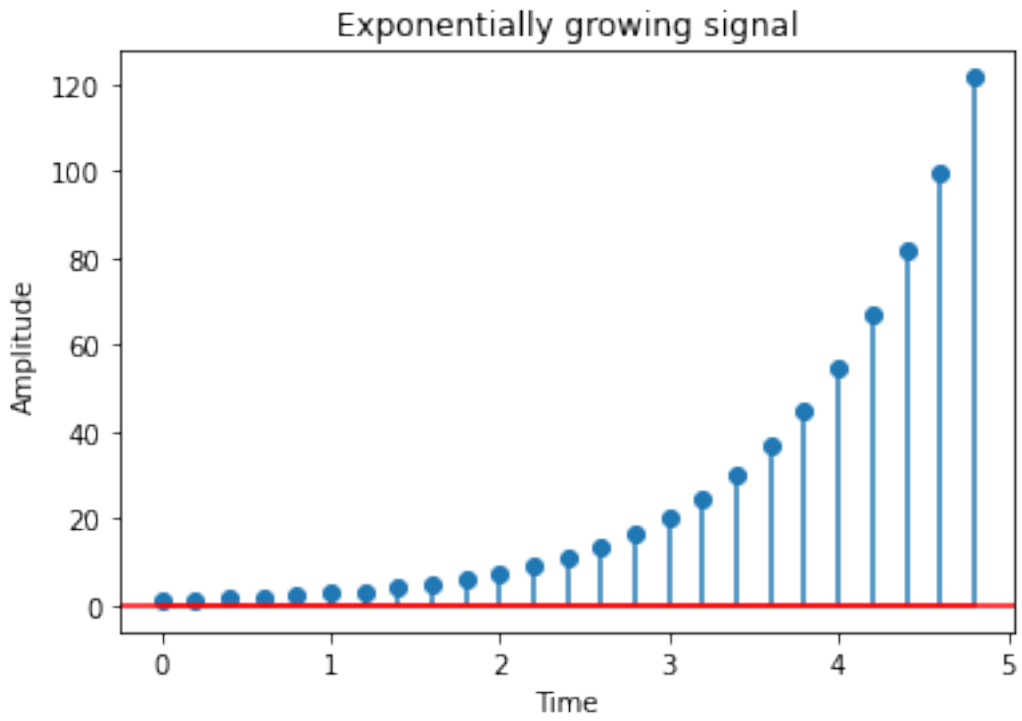
[8]:
```
# 2. Step signal

time = np.arange(-1,5,0.25)
y = list()
for t in time:
    if t>0:
        y.append(math.ceil(t))
    else:
        y.append(0)
amp = np.array(y)

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Step signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



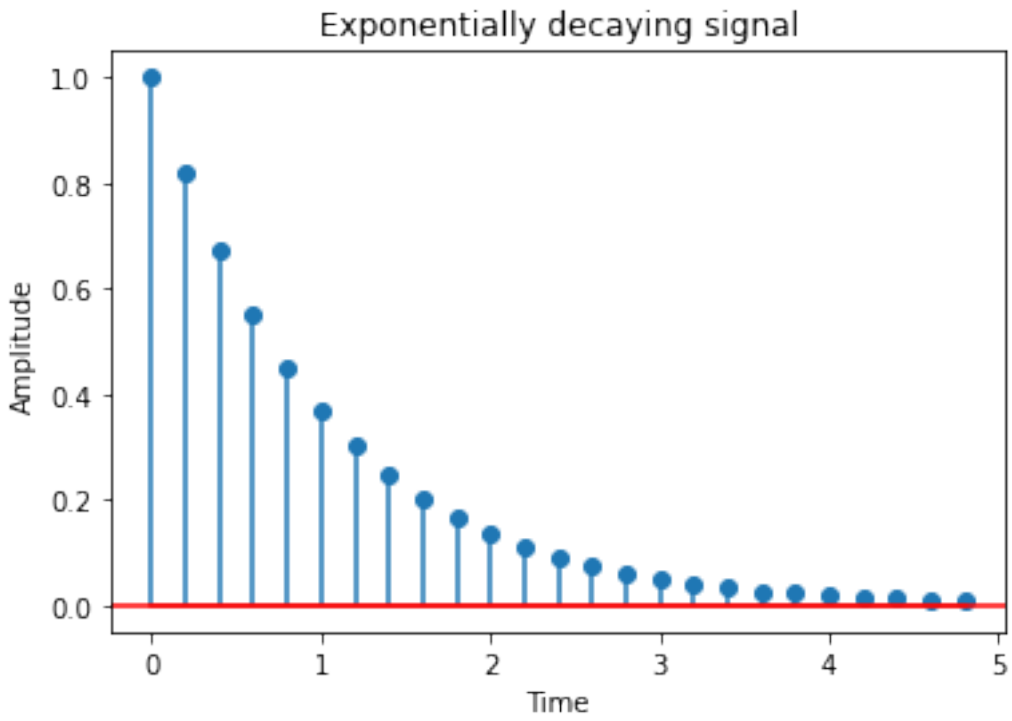
[9]:
```
# 3. Ramp signal

time = np.arange(-10,11,1)
amp = np.zeros(np.size(time))
index = np.where(time>=0)
amp[index] = time[index]

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Ramp signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



[10]:
```
# 4. Exponentially growing signal

time = np.arange(0,5,0.2)
amp = np.exp(time)

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Exponentially growing signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



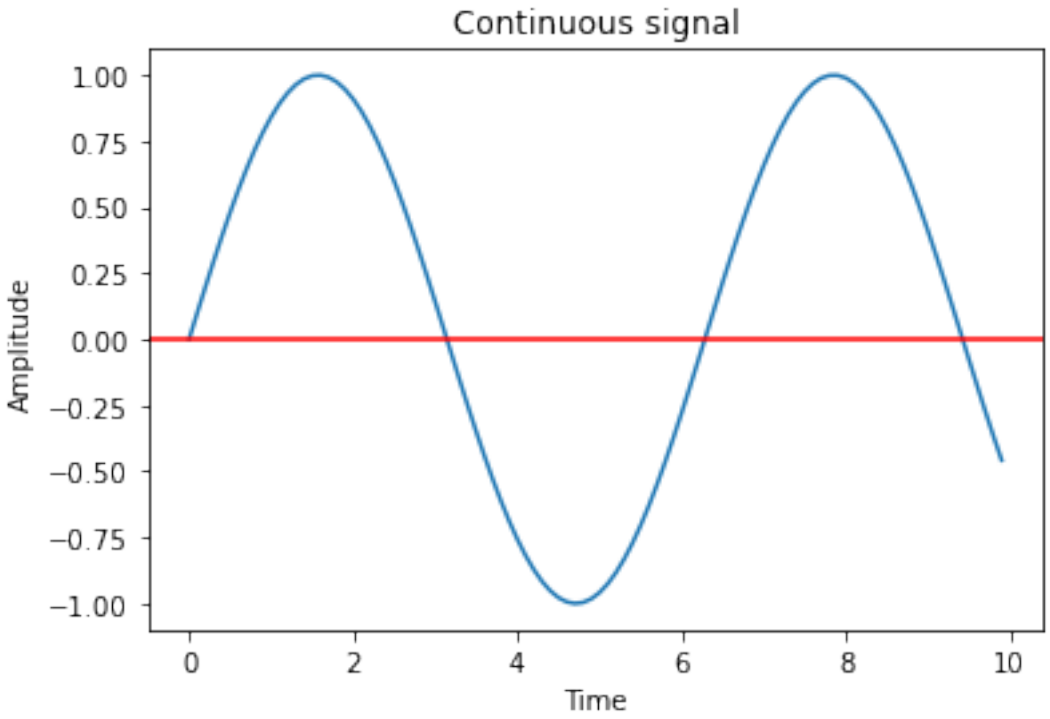
[11]:
```
# 5. Exponentially decaying signal

time = np.arange(0,5,0.2)
amp = np.exp(-time)

plt.figure()
plt.stem(time, amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Exponentially decaying signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

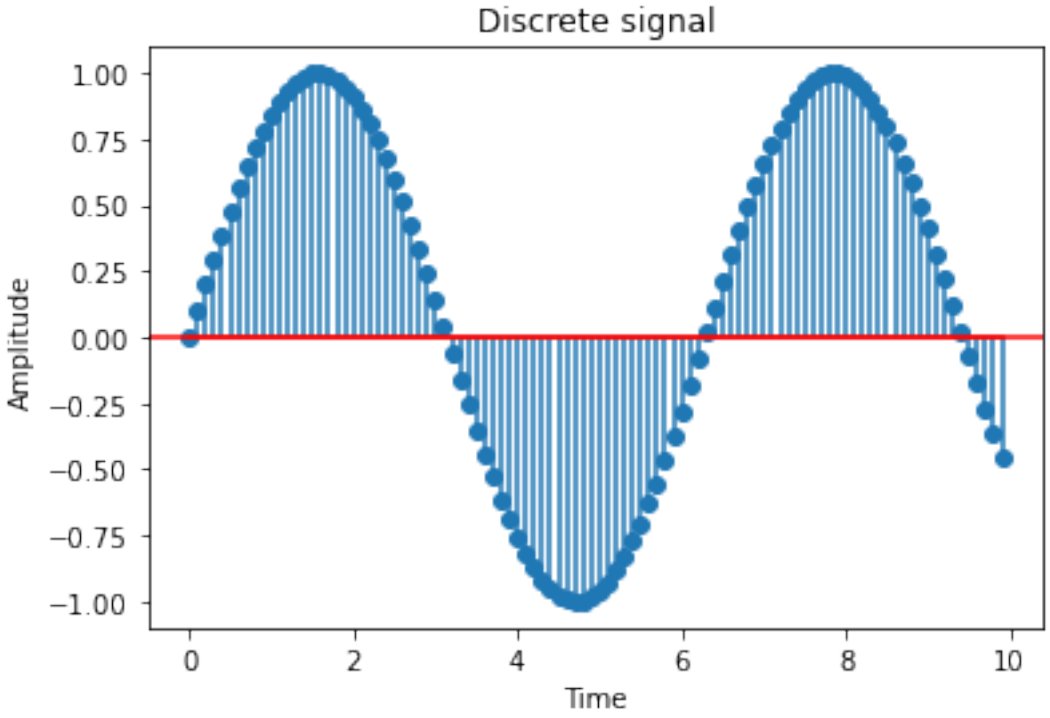


1.3 Question 3

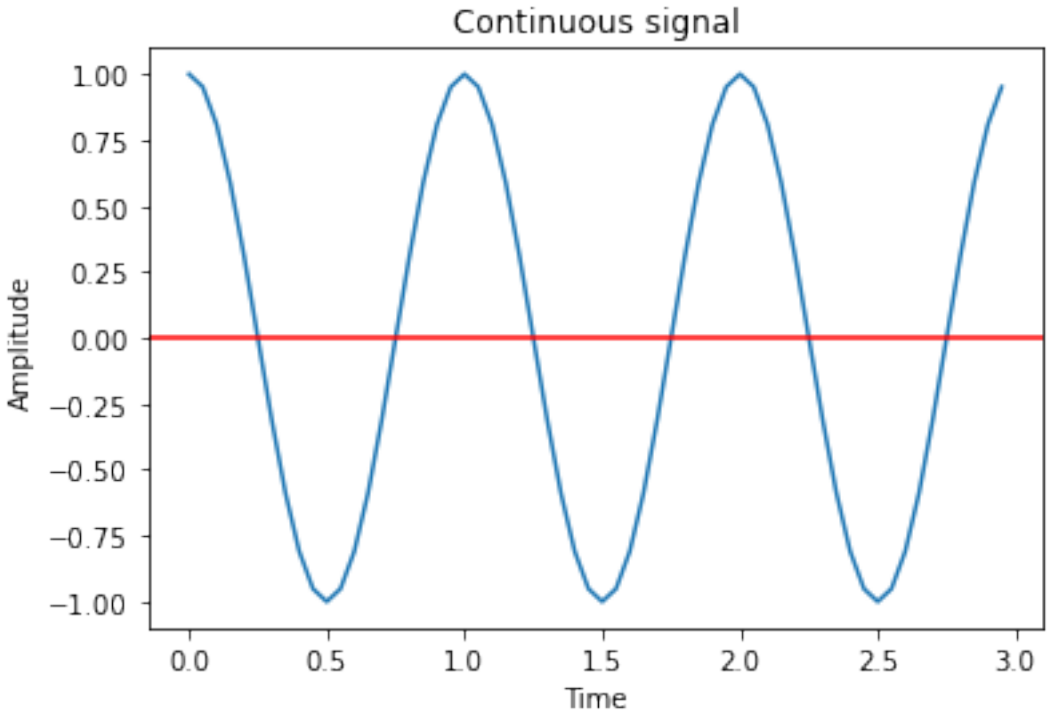
Generate continuous time and discrete time signal for the following

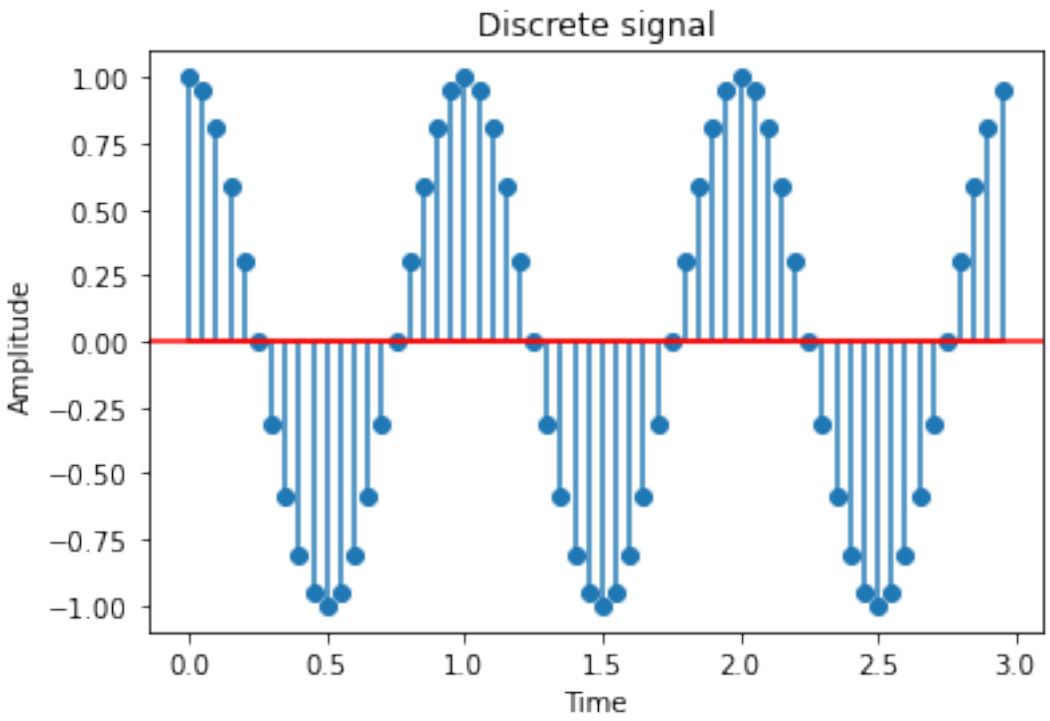
[12]:
```
# 1. y = sin(t)

time = np.arange(0,10,0.1)
amp = np.sin(time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Continuous signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Discrete signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```


Continuous signal
Amplitude
Time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10

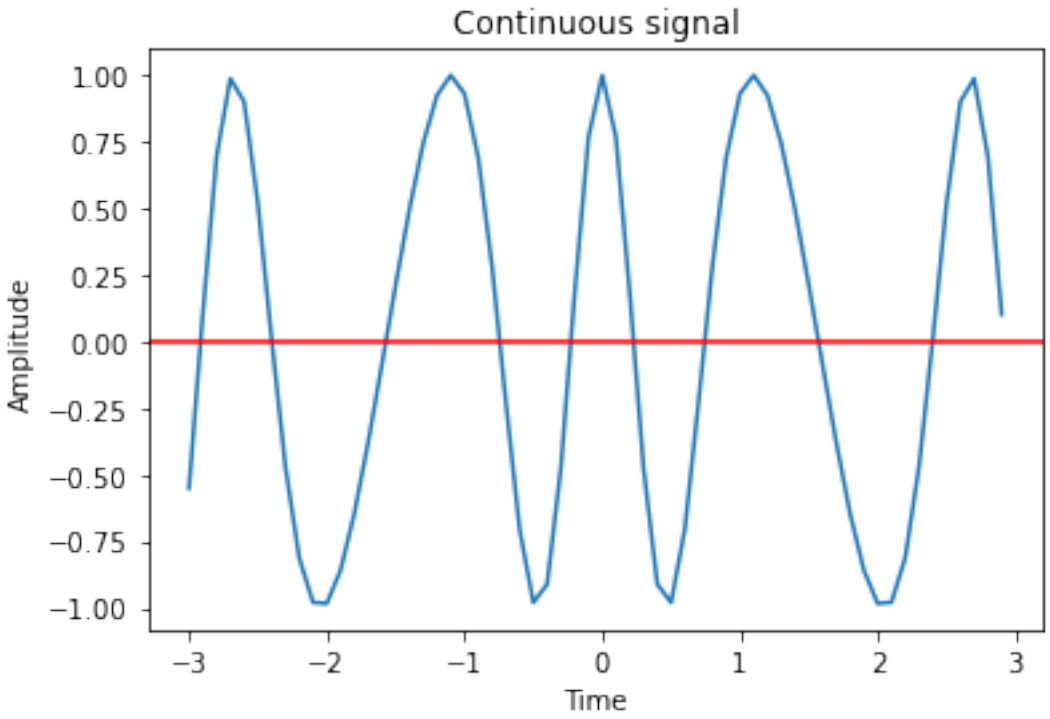

Discrete signal
Amplitude
Time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10

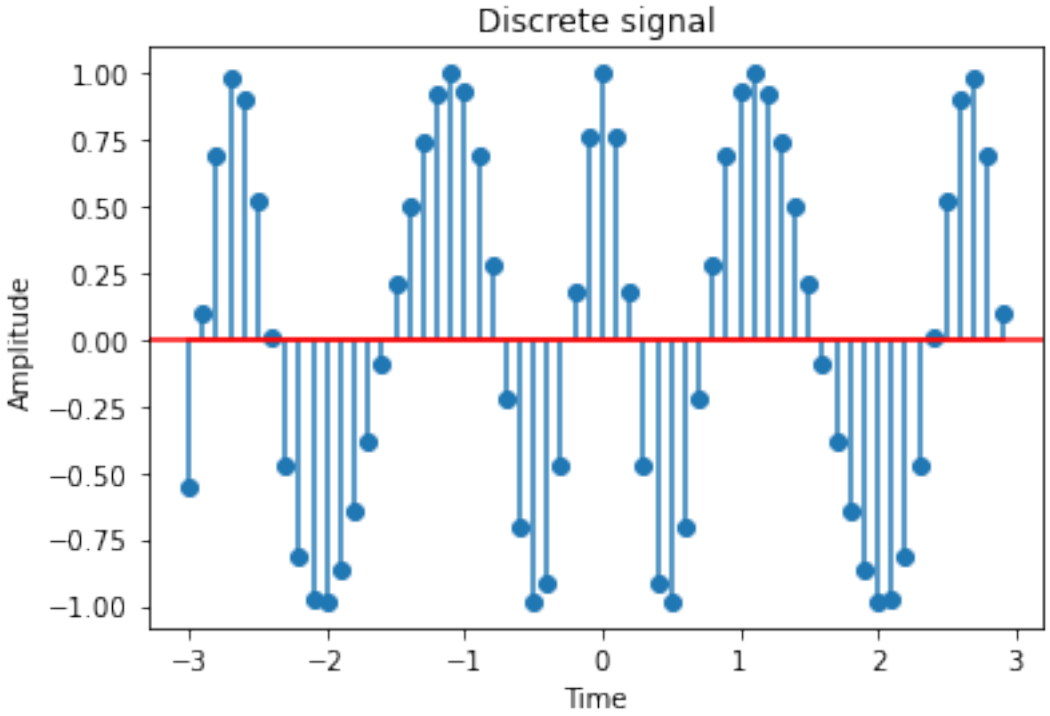
[13]:
```
# 2. y = cos(2 t)

time = np.arange(0,3,0.05)
amp = np.cos(2*np.pi*time)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Continuous signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Discrete signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```





[14]:
```
# 3. y = cos(5t + sin(2t))

time = np.arange(-3,3,0.1)
amp = np.cos(5*time + np.sin(2*time))

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.title('Continuous signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,amp,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('Discrete signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```


Continuous signal
Amplitude
Time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−3
−2
−1
0
1
2
3


Discrete signal
Amplitude
Time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−3
−2
−1
0
1
2
3

1.4 Question 4

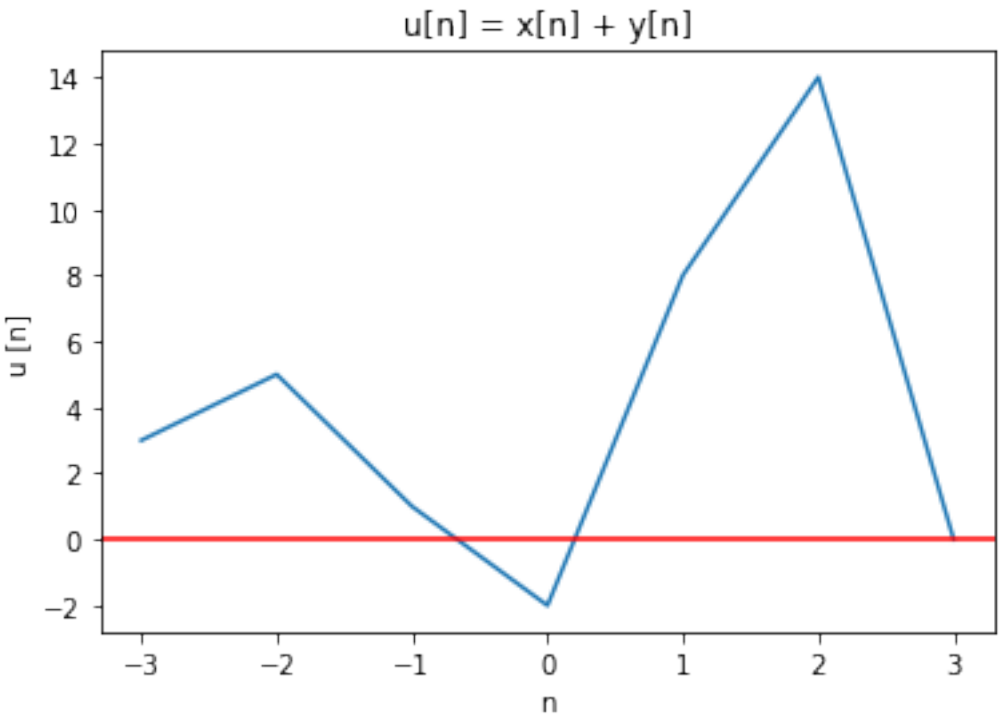
Consider the following length 7 sequences defined for -3<=n<=3

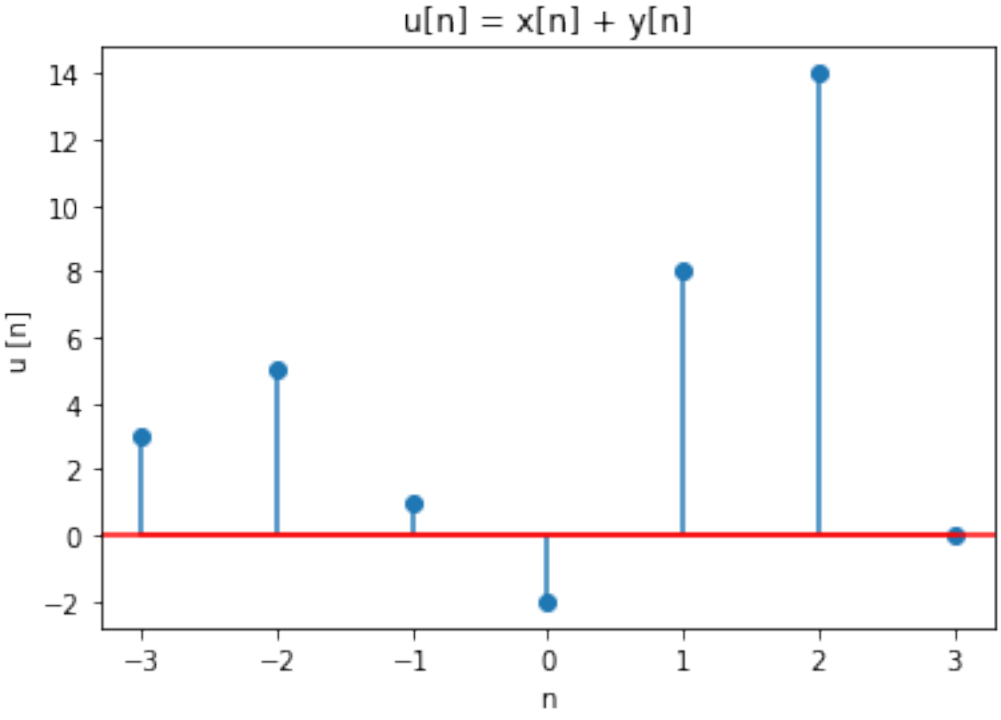
x[n] = [3, -2, 0, 1, 4, 5, 2] y[n] = [0, 7, 1,-3, 4, 9, -2] w[n] = [-5, 4, 3, 6, -5, 0, 1]

Generate the following sequences

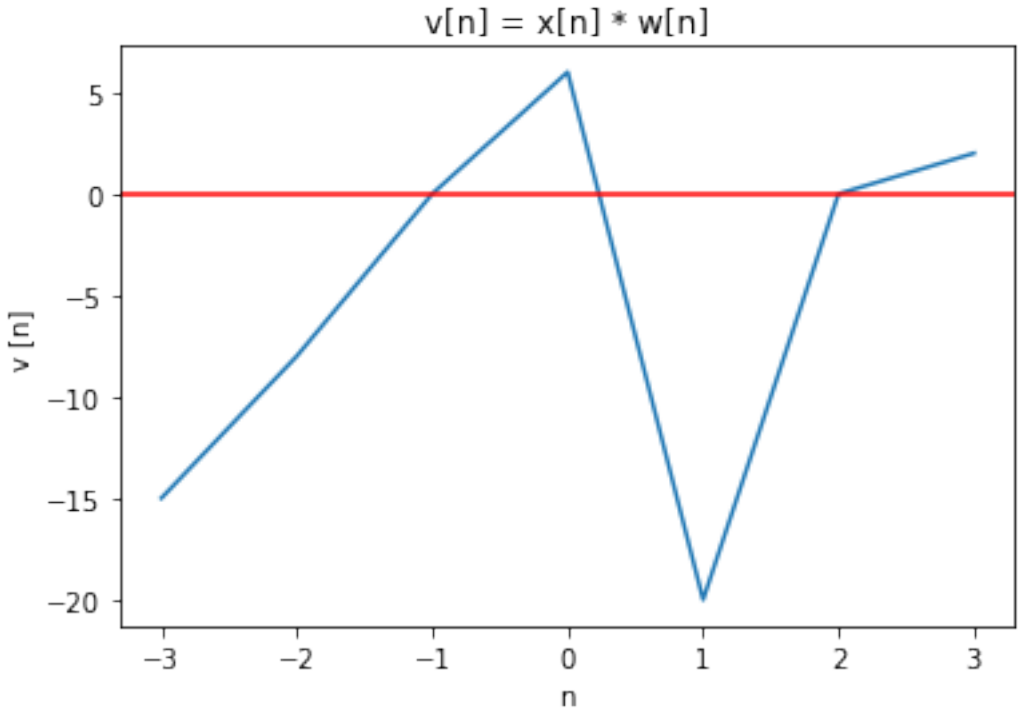
[15]:
```
# Common sequences for Q4
n = np.arange(-3,4,1)
x = np.array([3,-2,0,1,4,5,2])
y = np.array([0,7,1,-3,4,9,-2])
w = np.array([-5,4,3,6,-5,0,1])
```

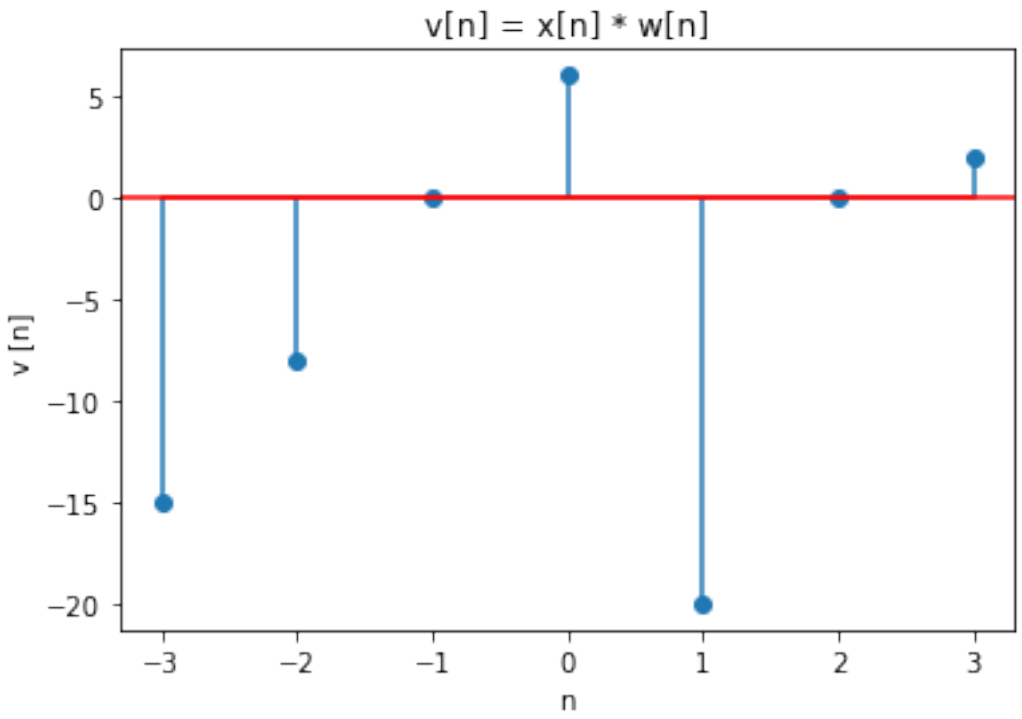
[16]:
```
# 1. u[n] = x[n] + y[n]

u = np.add(x,y)

plt.figure()
plt.plot(n,u)
plt.axhline(y=0, c='r')
plt.title('u[n] = x[n] + y[n]')
plt.xlabel('n')
plt.ylabel('u [n]')
plt.show()

plt.figure()
plt.stem(n,u,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('u[n] = x[n] + y[n]')
plt.xlabel('n')
plt.ylabel('u [n]')
plt.show()
```


u[n] = x[n] + y[n]
u [n]
n
14
12
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3

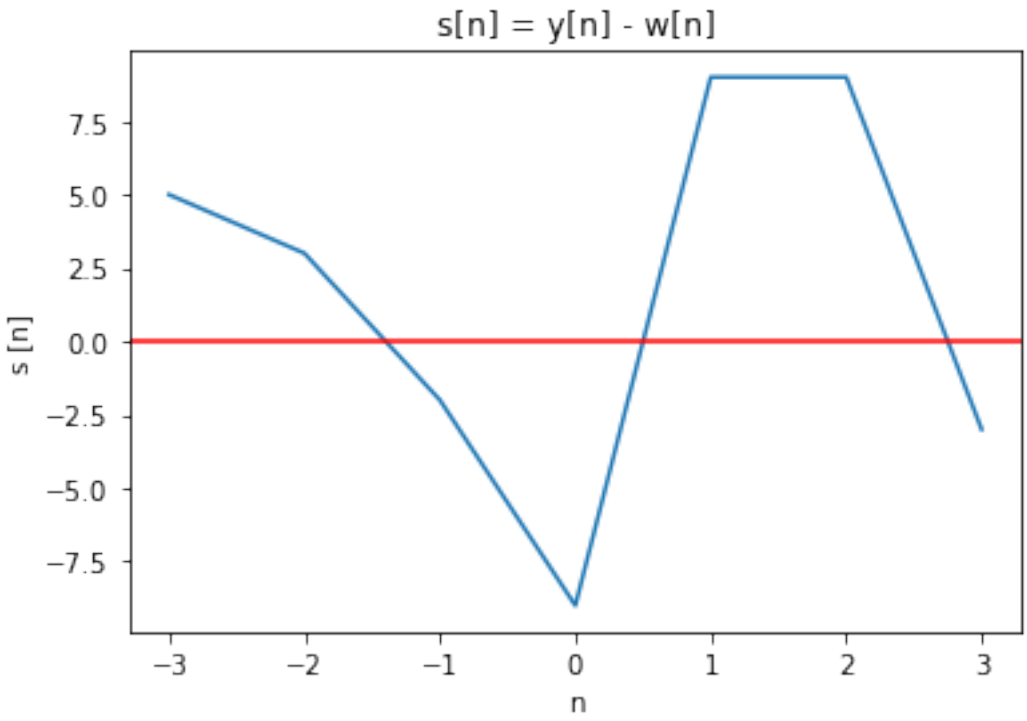

u[n] = x[n] + y[n]
u [n]
n
14
12
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3

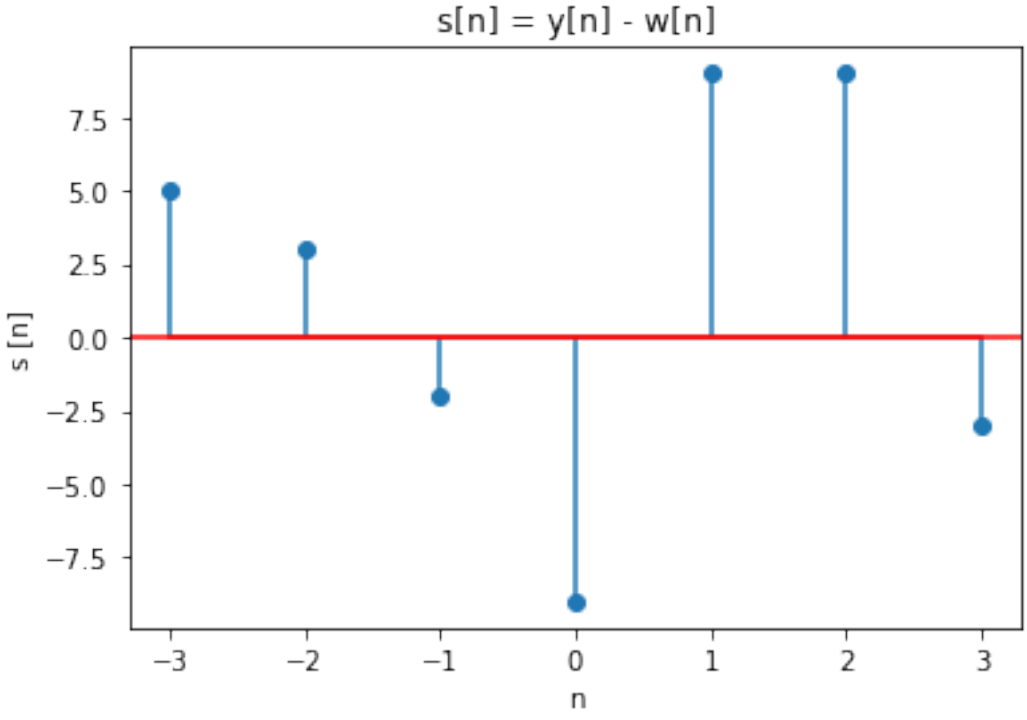
[17]:
```
# 2. v[n] = x[n] * w[n]

v = np.multiply(x,w)

plt.figure()
plt.plot(n,v)
plt.axhline(y=0, c='r')
plt.title('v[n] = x[n] * w[n]')
plt.xlabel('n')
plt.ylabel('v [n]')
plt.show()

plt.figure()
plt.stem(n,v,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('v[n] = x[n] * w[n]')
plt.xlabel('n')
plt.ylabel('v [n]')
plt.show()
```



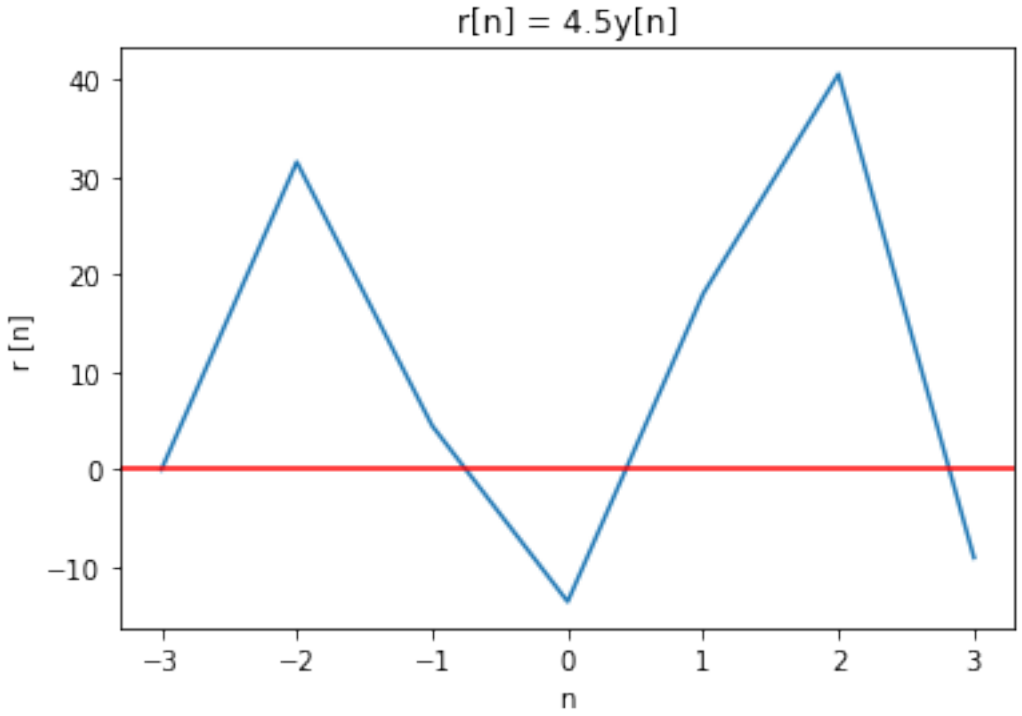


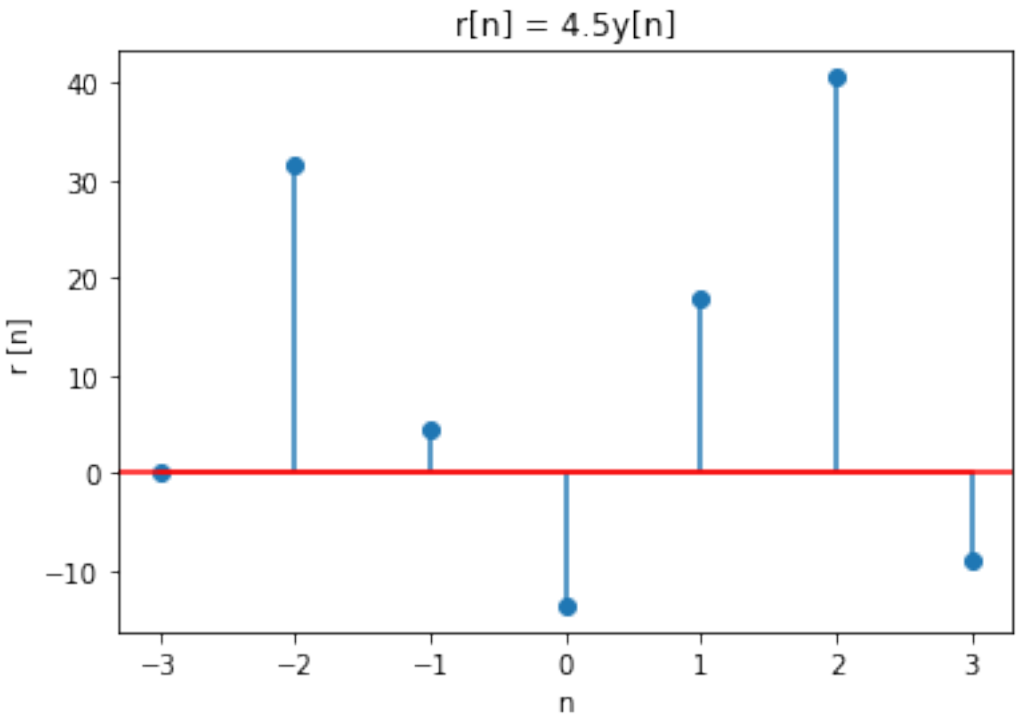
[18]:
```
# 3. s[n] = y[n] - w[n]

s = np.subtract(y,w)

plt.figure()
plt.plot(n,s)
plt.axhline(y=0, c='r')
plt.title('s[n] = y[n] - w[n]')
plt.xlabel('n')
plt.ylabel('s [n]')
plt.show()

plt.figure()
plt.stem(n,s,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('s[n] = y[n] - w[n]')
plt.xlabel('n')
plt.ylabel('s [n]')
plt.show()
```


s[n] = y[n] - w[n]
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
s [n]
−3
−2
−1
0
1
2
3
n


s[n] = y[n] - w[n]
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
s [n]
−3
−2
−1
0
1
2
3
n

[19]:
```
# 4. r[n] = 4.5y[n]

r = np.multiply(4.5,y)

plt.figure()
plt.plot(n,r)
plt.axhline(y=0, c='r')
plt.title('r[n] = 4.5y[n]')
plt.xlabel('n')
plt.ylabel('r [n]')
plt.show()

plt.figure()
plt.stem(n,r,use_line_collection=True)
plt.axhline(y=0, c='r')
plt.title('r[n] = 4.5y[n]')
plt.xlabel('n')
plt.ylabel('r [n]')
plt.show()
```



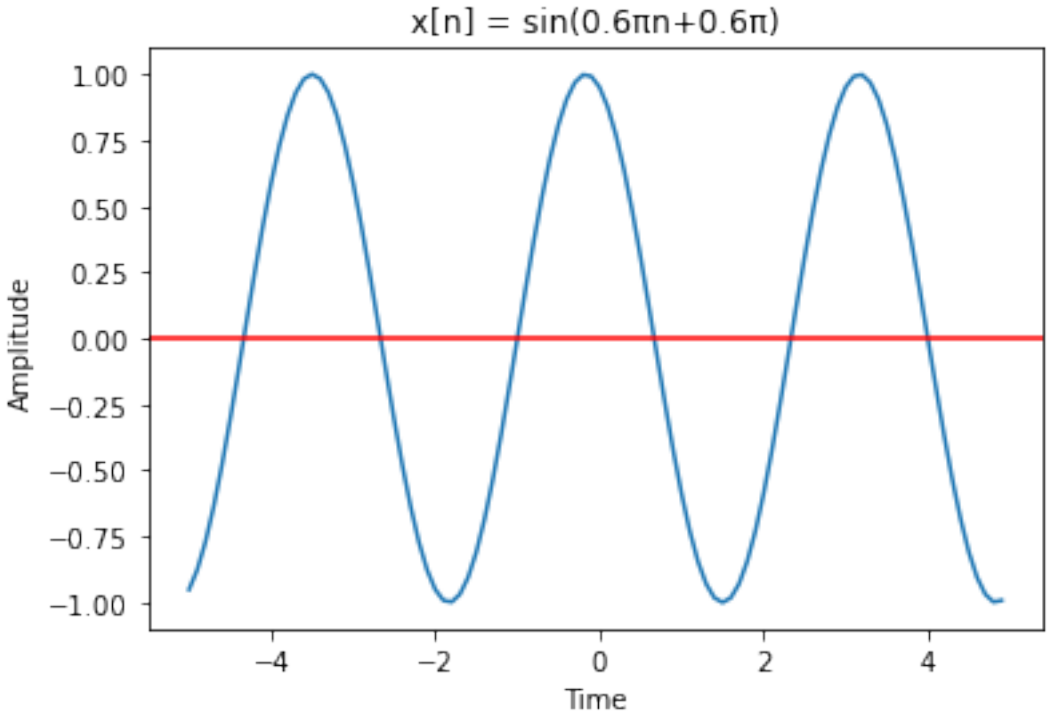


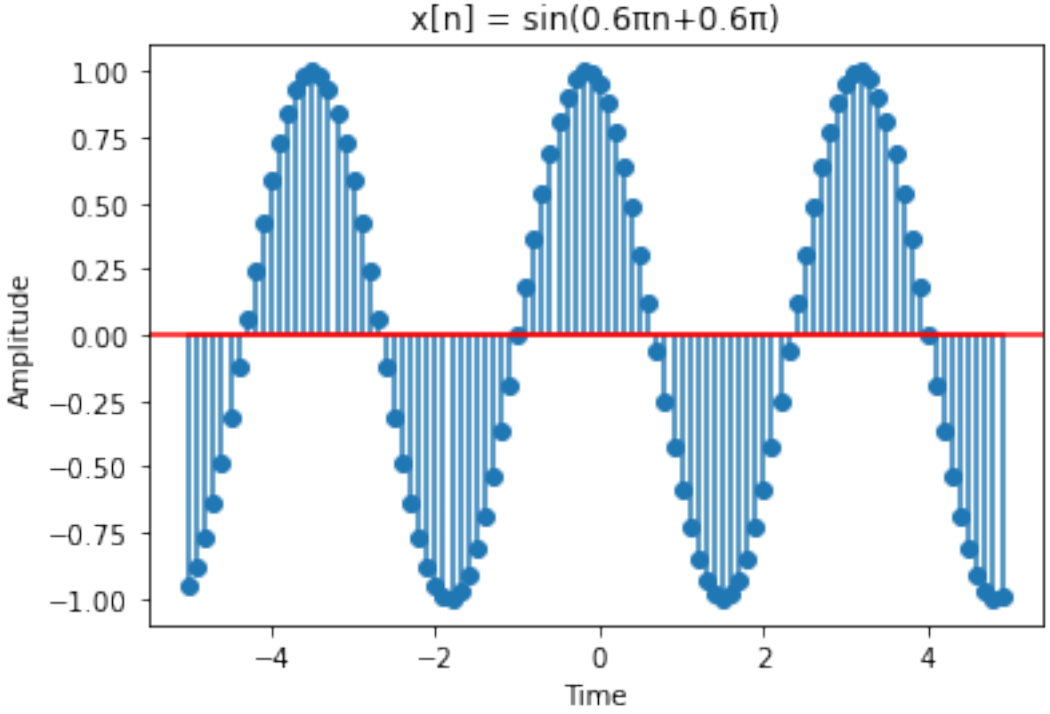
2 Question 5

Generate the sequences

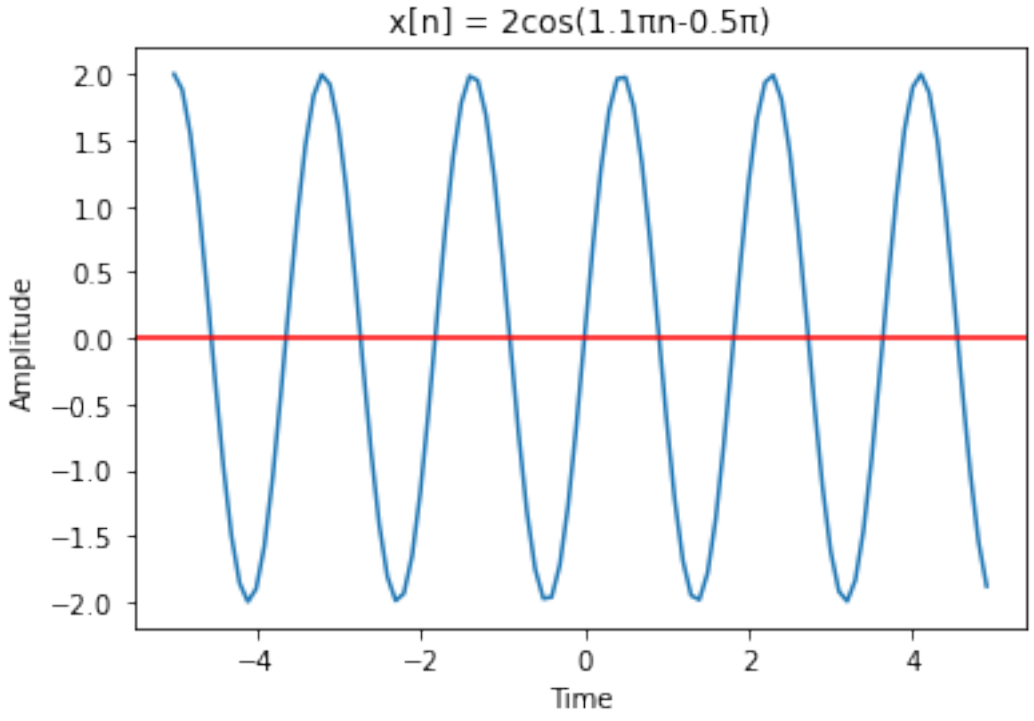
[20]:
```
# 1. x[n] = sin(0.6 n+0.6 )

time = np.arange(-5,5,0.1)
x = np.sin(0.6*np.pi*time + 0.6*np.pi)

plt.figure()
plt.plot(time,x)
plt.axhline(y=0, color='r')
plt.title('x[n] = sin(0.6 n+0.6 )')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,x,use_line_collection=True)
plt.axhline(y=0, color='r')
plt.title('x[n] = sin(0.6 n+0.6 )')
plt.xlabel('Time')
plt.ylabel('Amplitude')
```

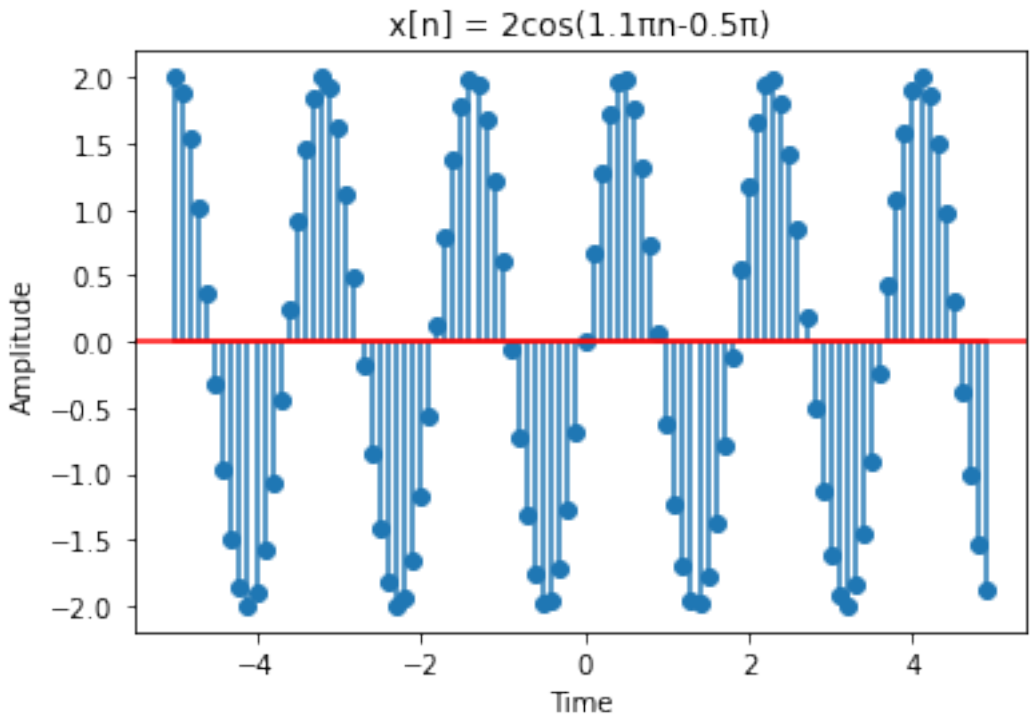
```
plt.show()
```


x[n] = sin(0.6πn+0.6π)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Amplitude
−4
−2
0
2
4
Time

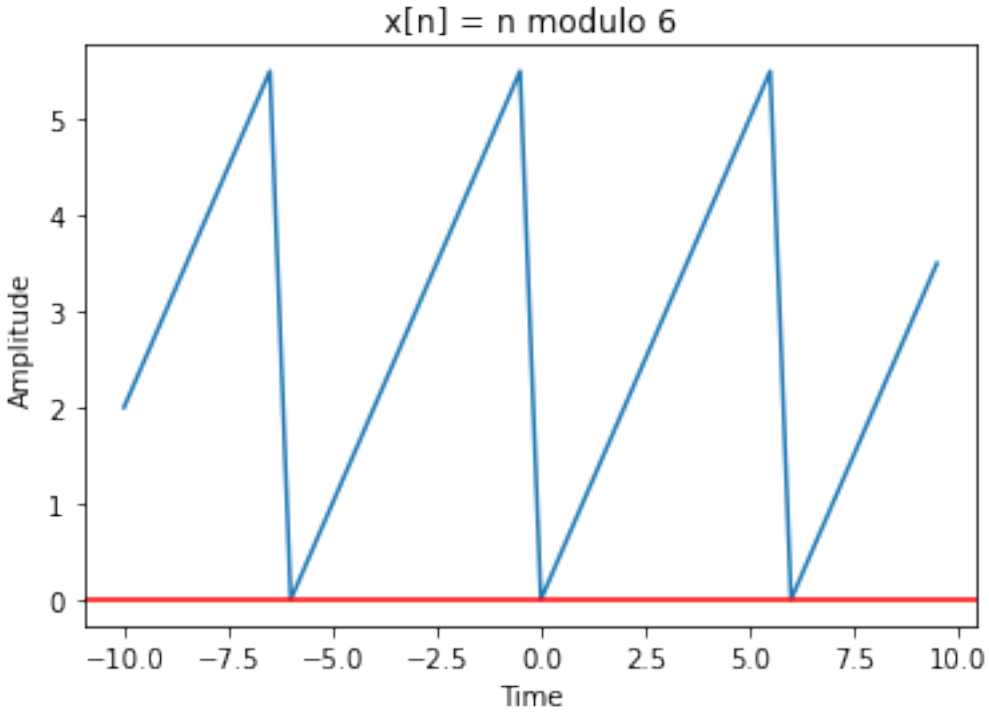

x[n] = sin(0.6πn+0.6π)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Amplitude
−4
−2
0
2
4
Time

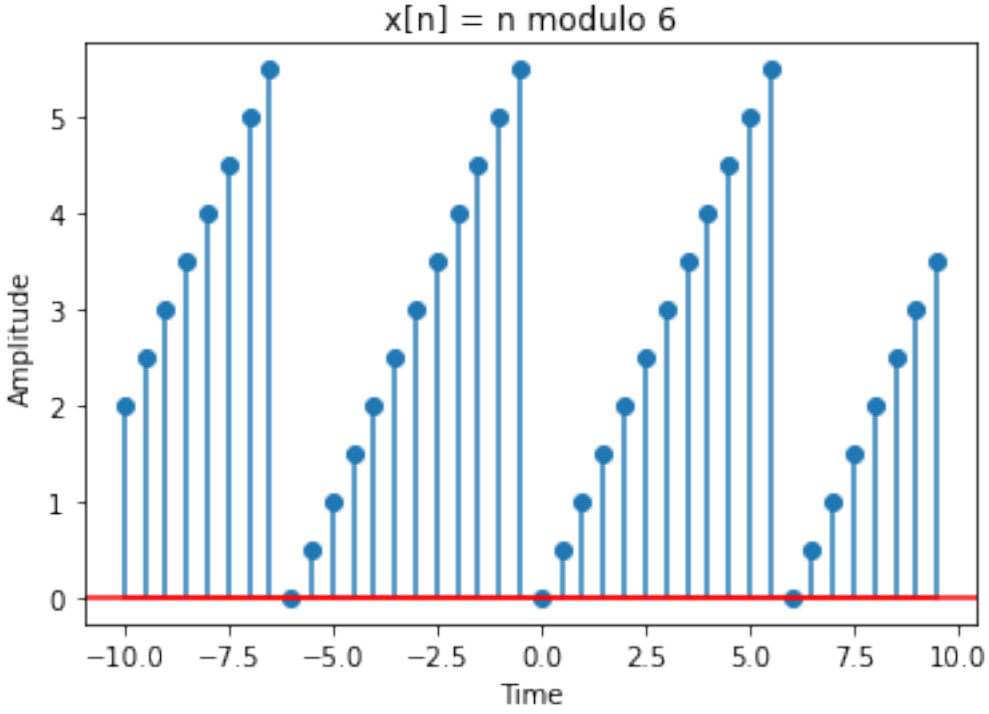
[21]:
```
# 2. x[n] = 2cos(1.1 n-0.5 )

time = np.arange(-5,5,0.1)
x = 2*np.cos(1.1*np.pi*time - 0.5*np.pi)

plt.figure()
plt.plot(time,x)
plt.axhline(y=0, color='r')
plt.title('x[n] = 2cos(1.1 n-0.5 )')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,x,use_line_collection=True)
plt.axhline(y=0, color='r')
plt.title('x[n] = 2cos(1.1 n-0.5 )')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```





[22]:
```
# 3. x[n] = n modulo 6

time = np.arange(-10,10,0.5)
x = np.mod(time,6)

plt.figure()
plt.plot(time,x)
plt.axhline(y=0, color='r')
plt.title('x[n] = n modulo 6')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.stem(time,x,use_line_collection=True)
plt.axhline(y=0, color='r')
plt.title('x[n] = n modulo 6')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```


x[n] = n modulo 6
Amplitude
Time
0
1
2
3
4
5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


x[n] = n modulo 6
Amplitude
Time
0
1
2
3
4
5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0