110704010 周世恩 programming hw1

Code explanation:

First, we define the signals.

```python
# Define the signals
def signal_1(n):
    if 1 <= n <= 20:
        return n
    elif 21 <= n <= 39:
        return 40 - n
    else:
        return 0

def signal_2(n):
    if 1 <= n <= 10:
        return 1
    else:
        return 0
```

Then we plot it out for (a)

```python
# (a) Plot signals
n_values = np.arange(1, 40)
signal1_values = [signal_1(n) for n in n_values]
signal2_values = [signal_2(n) for n in n_values]

plt.stem(n_values, signal1_values, label='Signal 1')
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.title('Plot of Signal 1')
plt.legend()
plt.show()

plt.stem(n_values, signal2_values, label='Signal 2')
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.title('Plot of Signal 2')
plt.legend()
plt.show()
```

For (b), we convolve two signals with np.convolve() and plot the result out.

```
# (b) Convolve signals using numpy.convolve
conv_result = np.convolve(signal1_values, signal2_values, mode='full')

plt.stem(np.arange(1, len(conv_result)+1), conv_result)
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.title('Convolution using numpy.convolve')
plt.show()
```

For (c), we define a convolution function using matrix form and use it to convolve two signals. Then, we plot the result out.

```
# (c) Convolve signals using matrix form
def convolution_matrix(signal1, signal2):
    len1 = len(signal1)
    len2 = len(signal2)
    result_length = len1 + len2 - 1
    convolution_matrix = np.zeros((result_length, len1))

    for i in range(result_length):
        for j in range(len1):
            if i - j >= 0 and i - j < len2:
                convolution_matrix[i, j] = signal2[i - j]

    return np.dot(convolution_matrix, signal1)

conv_result_matrix = convolution_matrix(signal1_values, signal2_values)

plt.stem(np.arange(1, len(conv_result_matrix)+1), conv_result_matrix)
plt.xlabel('n')
plt.ylabel('Amplitude')
plt.title('Convolution using matrix form')
plt.show()
```

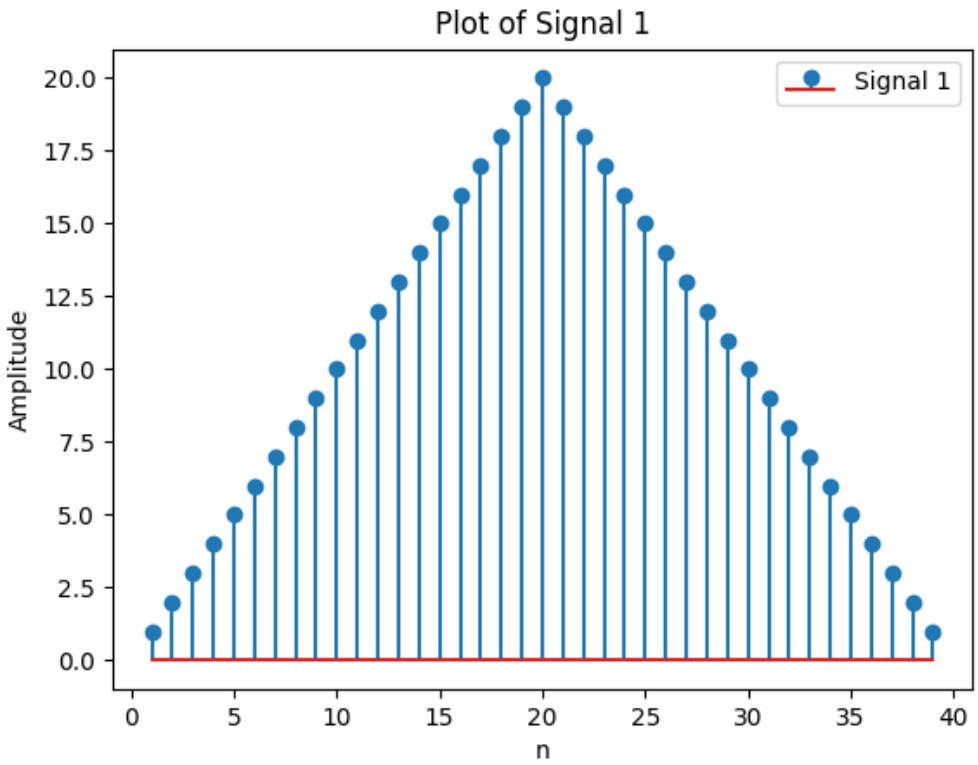
For (d), we define two new signals and do the same thing as above.

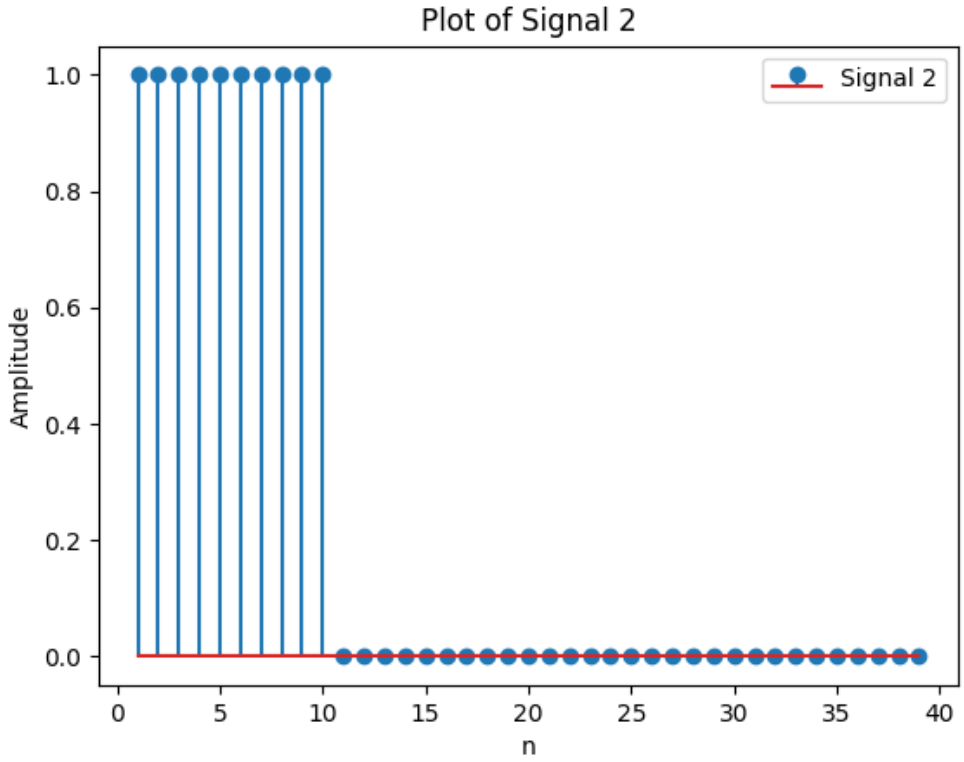
```python
# (d) using different signals
def sig_1(n):
    if 1 <= n <= 3:
        return 3 ** n
    else:
        return 0

def sig_2(n):
    if 1 <= n <= 5:
        return 2 ** n
    else:
        return 0
```

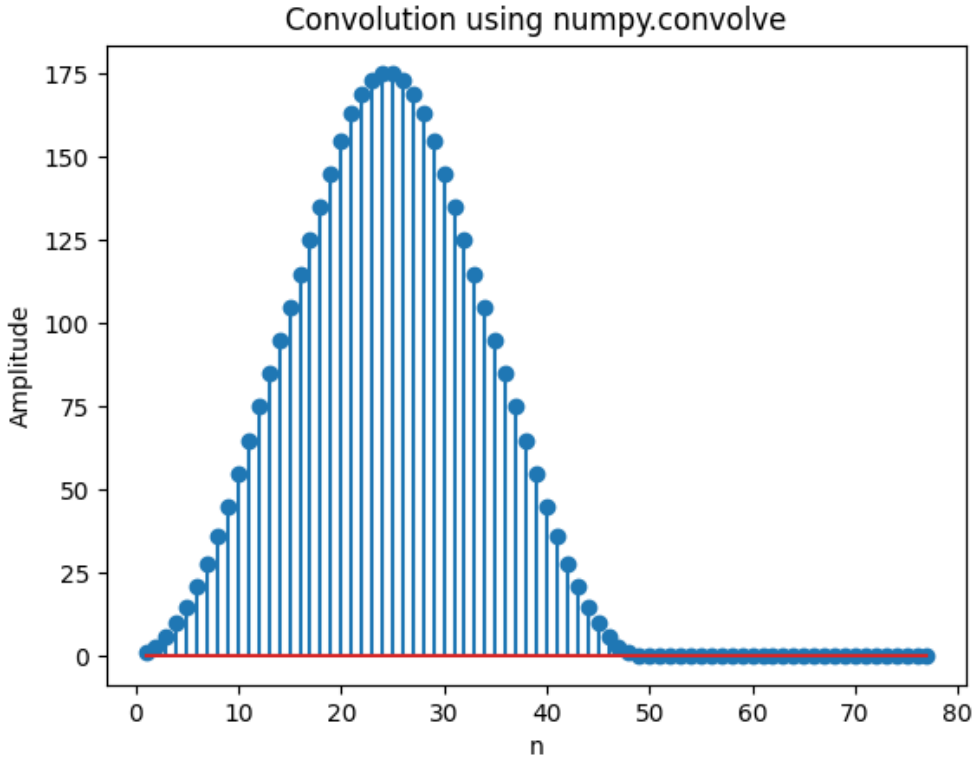
Figures:

(a)

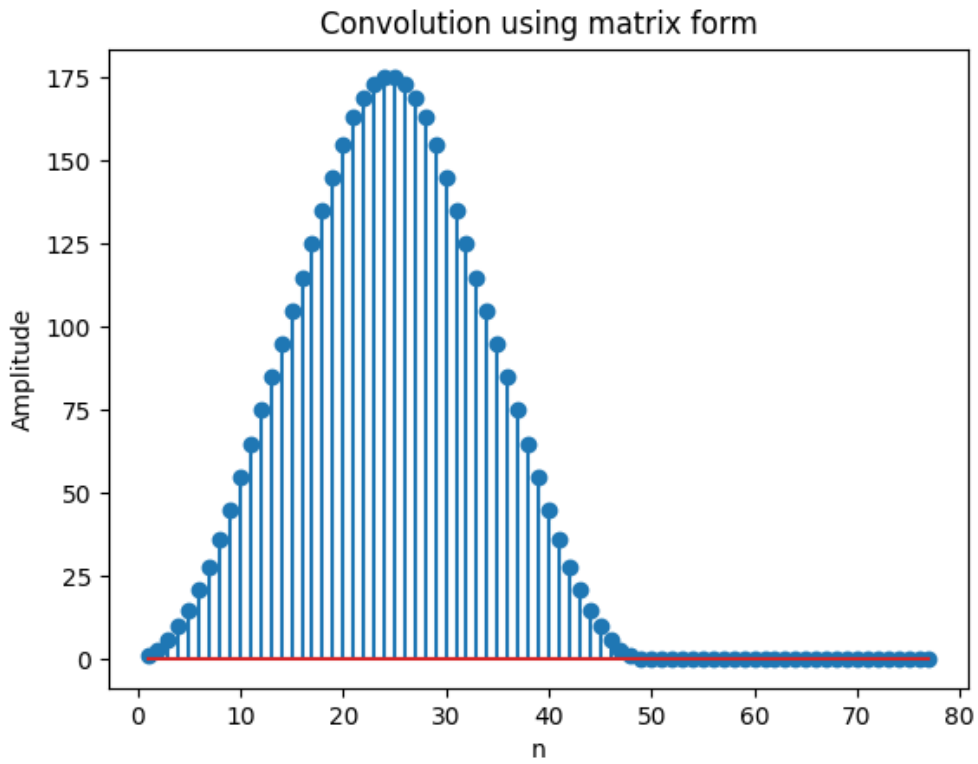



Plot of Signal 2
Signal 2
Amplitude
n
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35
40

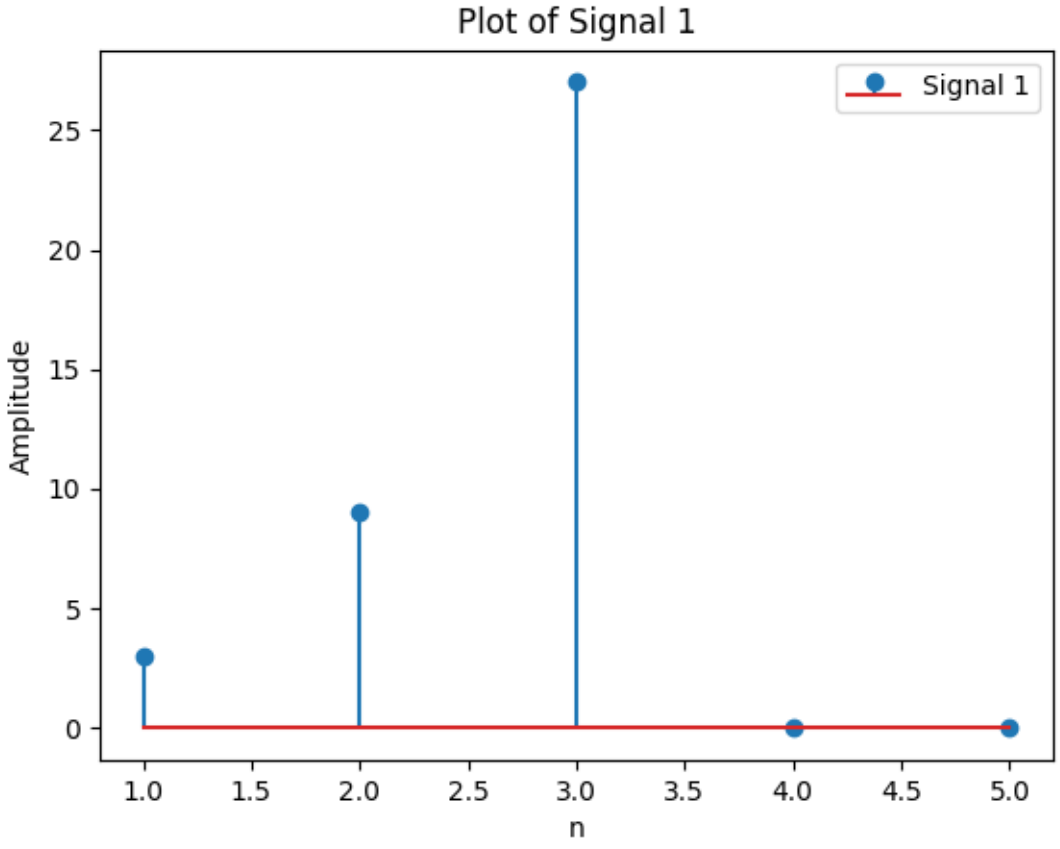
(b)

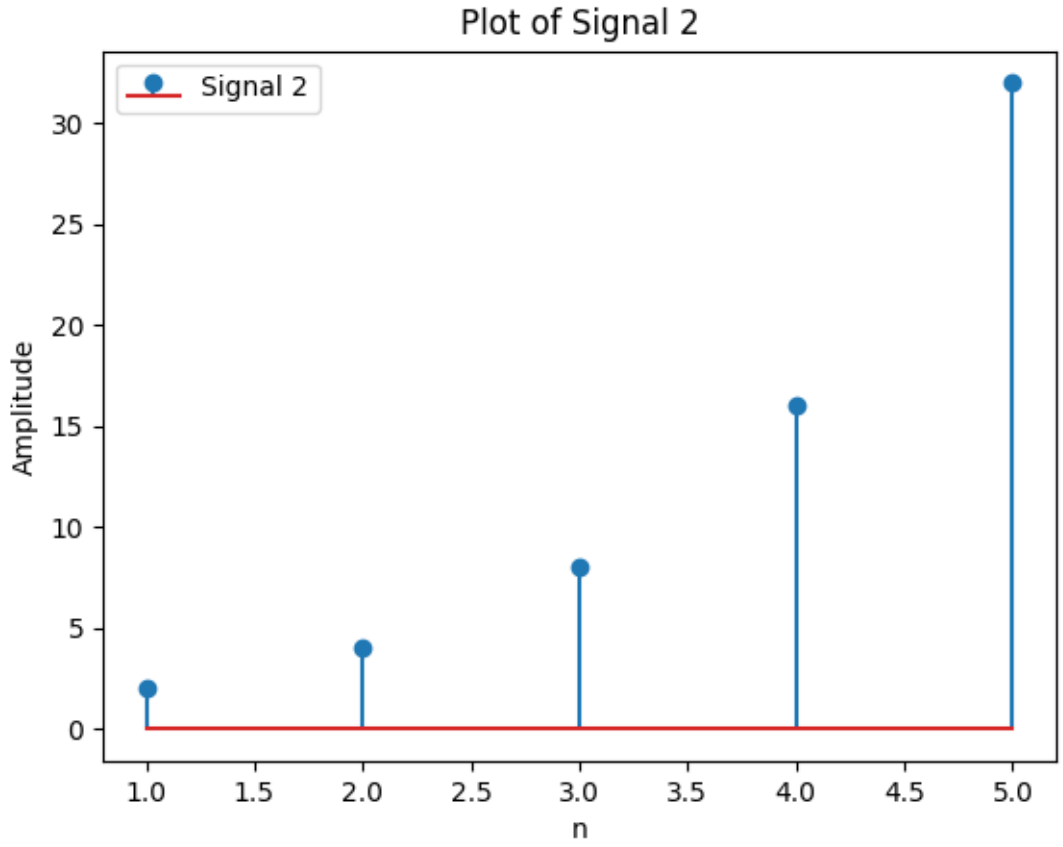

Convolution using numpy.convolve
Amplitude
n
175
150
125
100
75
50
25
0
0
10
20
30
40
50
60
70
80

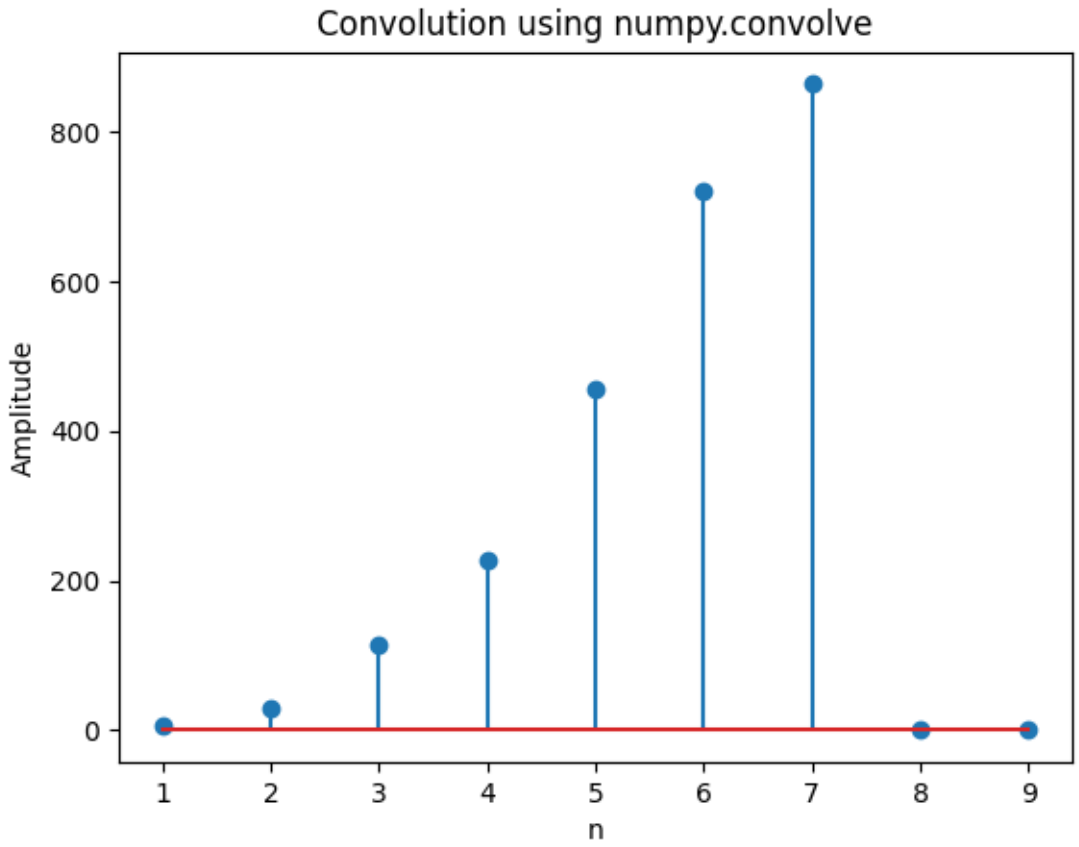
(c)

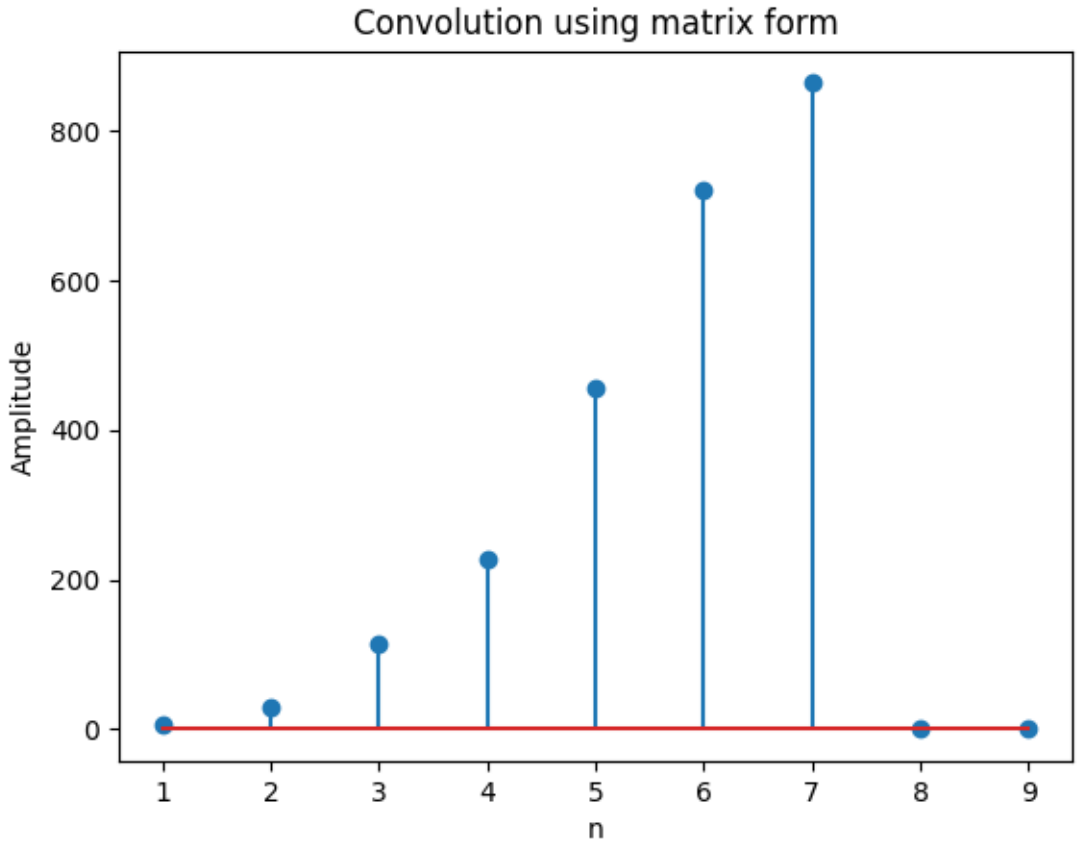

Convolution using matrix form
Amplitude
n
0
25
50
75
100
125
150
175
0
10
20
30
40
50
60
70
80

(d)


Plot of Signal 1
Signal 1
Amplitude
n
0
5
10
15
20
25
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Plot of Signal 2
Signal 2
Amplitude
n
0
5
10
15
20
25
30
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Convolution using numpy.convolve
Amplitude
n
0
200
400
600
800
1
2
3
4
5
6
7
8
9


Convolution using matrix form
Amplitude
800
600
400
200
0
1
2
3
4
5
6
7
8
9
n