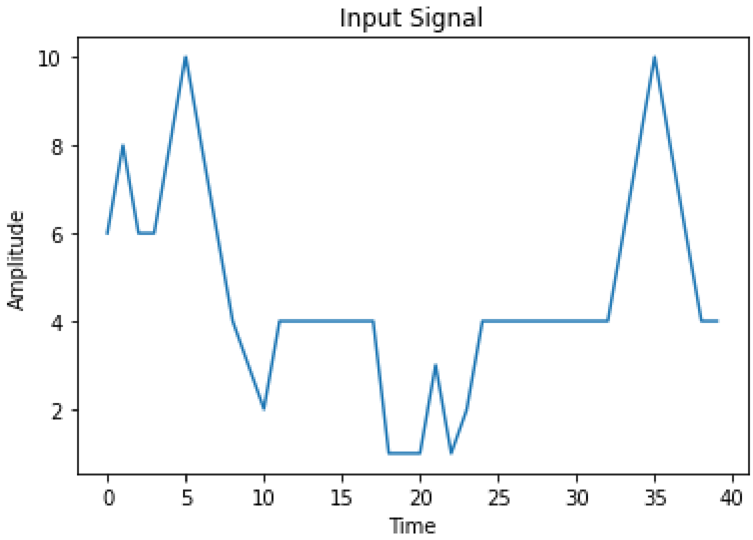
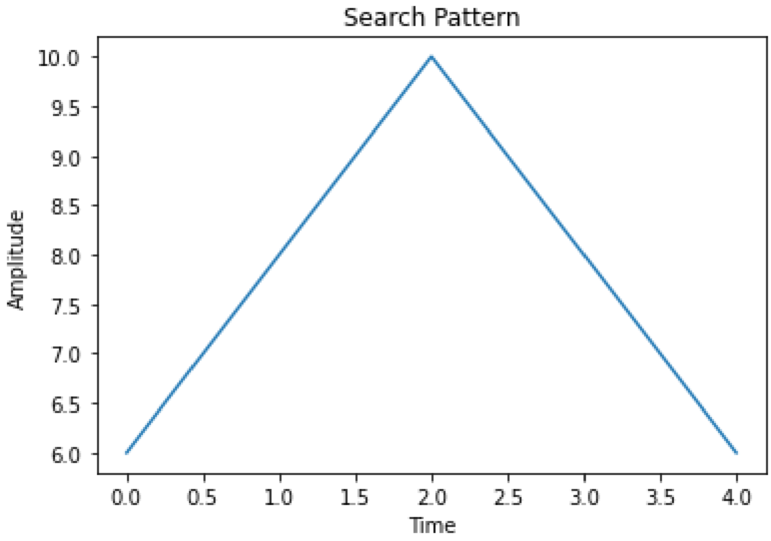
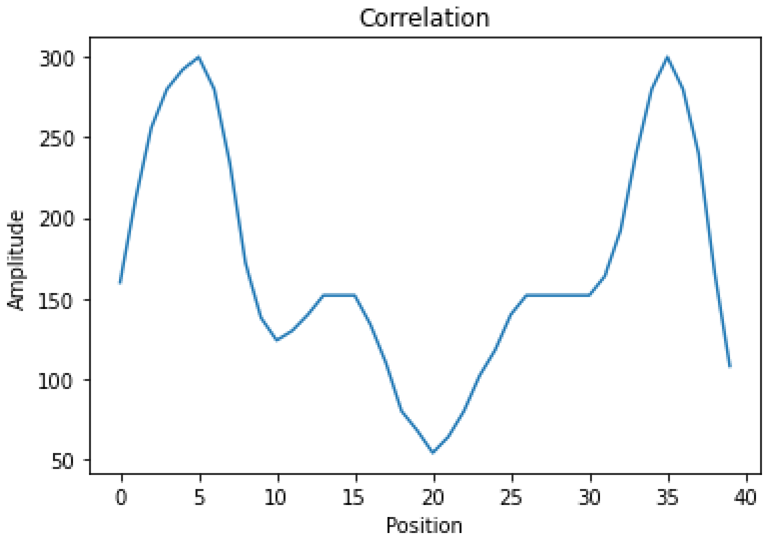
In [9]:

```
#Q1   NORMAL SIGNAL - SINGLE MATCH

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

answer = []
Input = np.array([6, 8, 6, 6, 8, 10, 8, 6, 4 ,3, 2, 4, 4, 4, 4, 4, 4, 4, 1, 1,1, 3, 1,
2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 8, 10, 8, 6, 4, 4])
search = np.array([6, 8, 10, 8, 6])
corr = signal.correlate(Input, search, mode = 'same')   #Correlation performed
length = len(corr)
Element = np.max(corr)
for i in range(0, length):
    if corr[i] == Element:
        answer.append(i - 1)
        break                    #We break because it is Single Matching, so only one m
atch is required

#Showing where the patterns have been
print('Pattern found at starting locations: ', answer)

#Plotting the Input Signal
plt.plot(Input)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Input Signal')
plt.show()

#Plotting the Pattern to be searched
plt.plot(search)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Search Pattern')
plt.show()

#Plotting the Correlation Signal
plt.plot(corr)
plt.xlabel('Position')
plt.ylabel('Amplitude')
plt.title('Correlation')
plt.show()
```

```
Pattern found at starting locations:  [4]
```



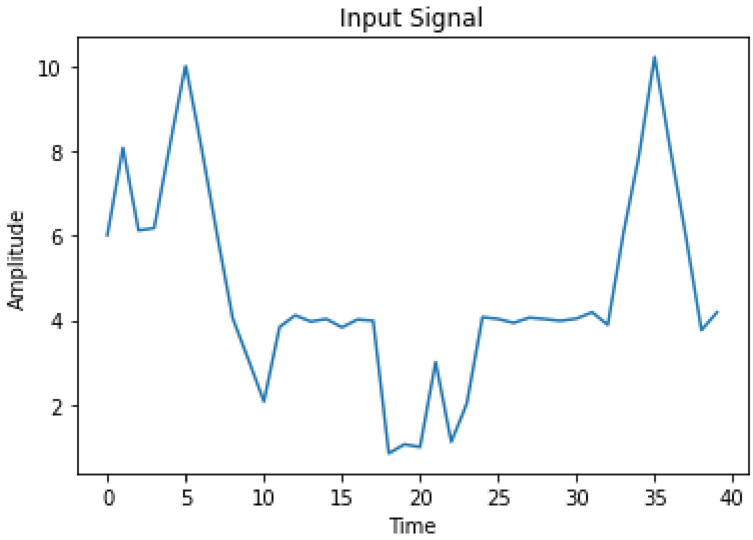


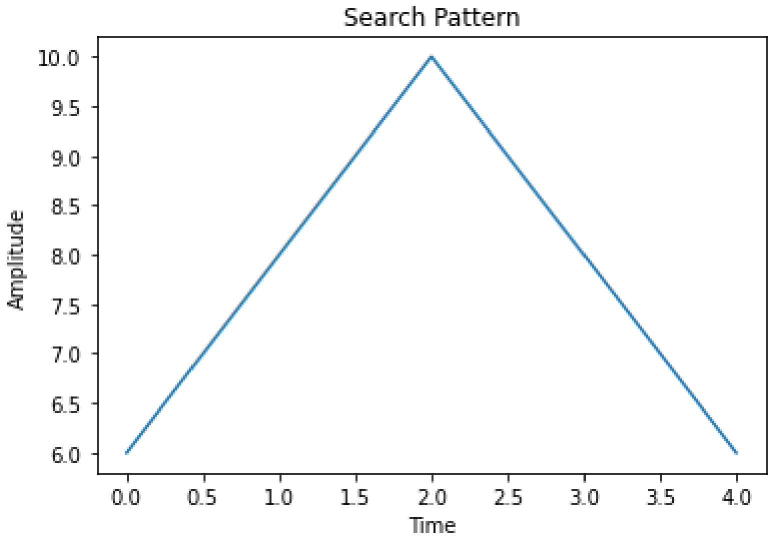


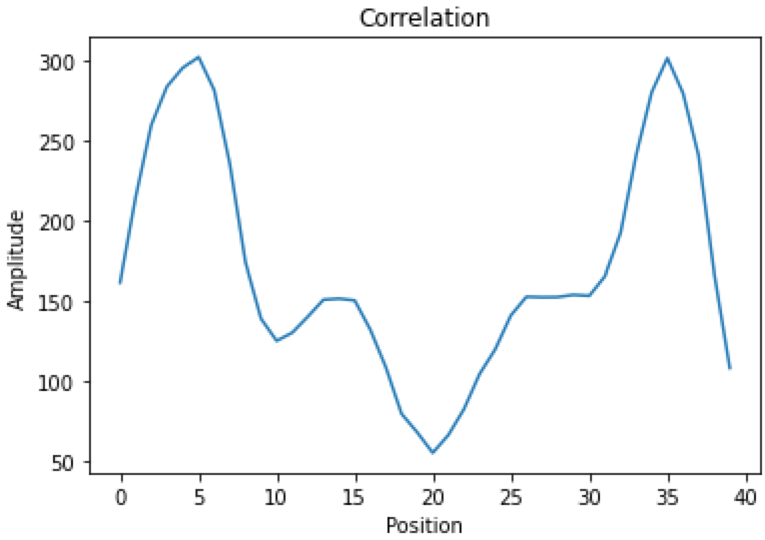
In [10]:

```
#Q2   NOISY SIGNAL - SINGLE MATCH

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

answer = []
Input = np.array([6, 8, 6, 6, 8, 10, 8, 6, 4 ,3, 2, 4, 4, 4, 4, 4, 4, 4, 1, 1,1, 3, 1,
2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 8, 10, 8, 6, 4, 4])
Input = Input + np.random.normal(0,0.1,len(Input))
search = np.array([6, 8, 10, 8, 6])
corr = signal.correlate(Input, search, mode = 'same')      #Correlation performed
length = len(corr)
Element = np.max(corr)
for i in range(0, length):
    if (corr[i] == Element):
        answer.append(i - 1)
        break                      #We break because it is Single Matching, so only one
match is required

#Showing where the patterns have been
print('Pattern found at starting locations: ', answer)

#Plotting the Input Signal
plt.plot(Input)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Input Signal')
plt.show()

#Plotting the Pattern to be searched
plt.plot(search)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Search Pattern')
plt.show()

#Plotting the Correlation Signal
plt.plot(corr)
plt.xlabel('Position')
plt.ylabel('Amplitude')
plt.title('Correlation')
plt.show()
```

```
Pattern found at starting locations:  [4]
```



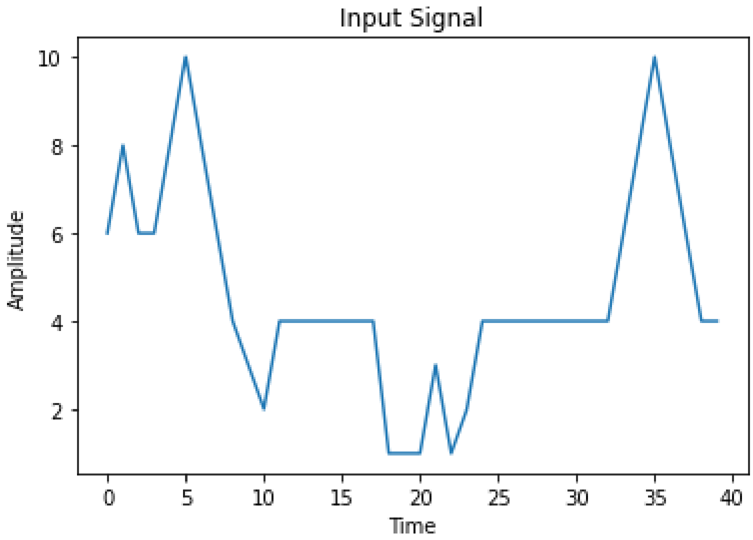


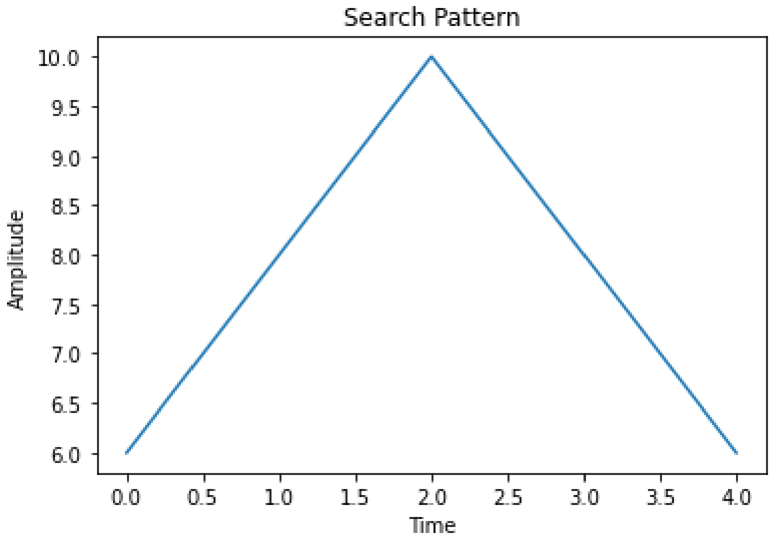


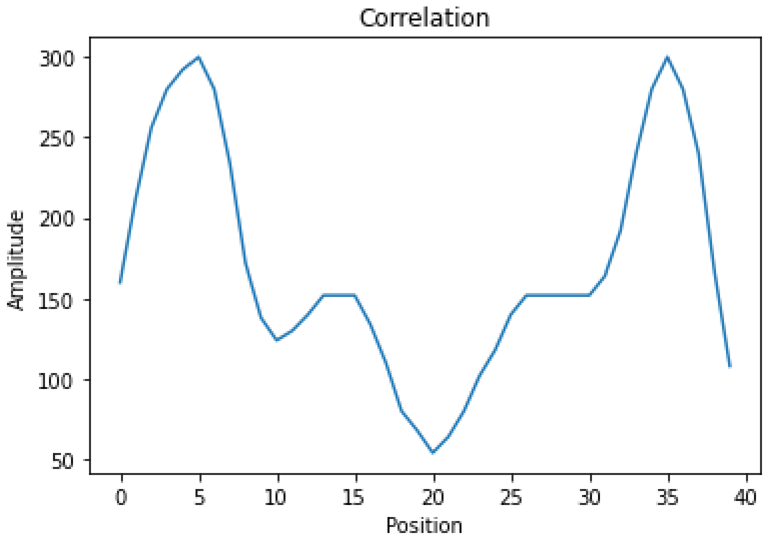
In [11]:

```
#Q3   NORMAL SIGNAL - MULTIPLE MATCH

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

answer = []
Input = np.array([6, 8, 6, 6, 8, 10, 8, 6, 4 ,3, 2, 4, 4, 4, 4, 4, 4, 4, 1, 1,1, 3, 1,
2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 8, 10, 8, 6, 4, 4])
search = np.array([6, 8, 10, 8, 6])
corr = signal.correlate(Input, search, mode = 'same')   #Correlation performed
length = len(corr)
Element = np.max(corr)
for i in range(0, length):
    if corr[i] == Element:
        answer.append(i - 1)       #All matches will be stored in the output

#Showing where the patterns have been
print('Pattern found at starting locations: ', answer)

#Plotting the Input Signal
plt.plot(Input)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Input Signal')
plt.show()

#Plotting the Pattern to be searched
plt.plot(search)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Search Pattern')
plt.show()

#Plotting the Correlation Signal
plt.plot(corr)
plt.xlabel('Position')
plt.ylabel('Amplitude')
plt.title('Correlation')
plt.show()
```

```
Pattern found at starting locations:  [4, 34]
```



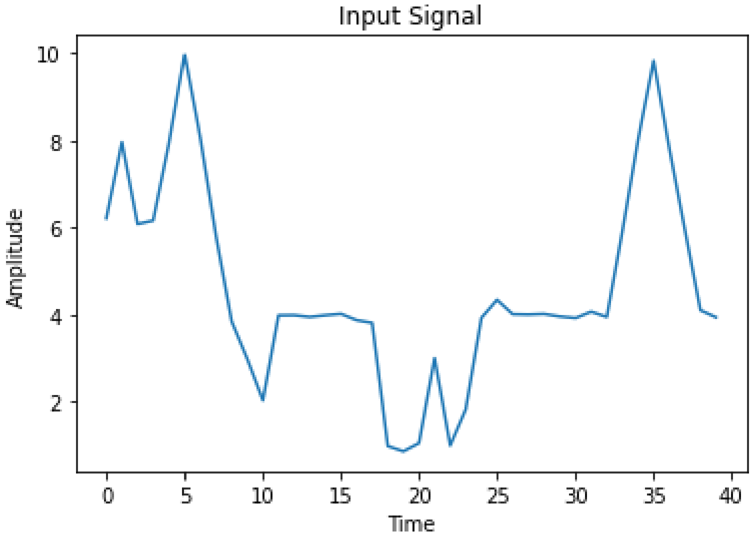


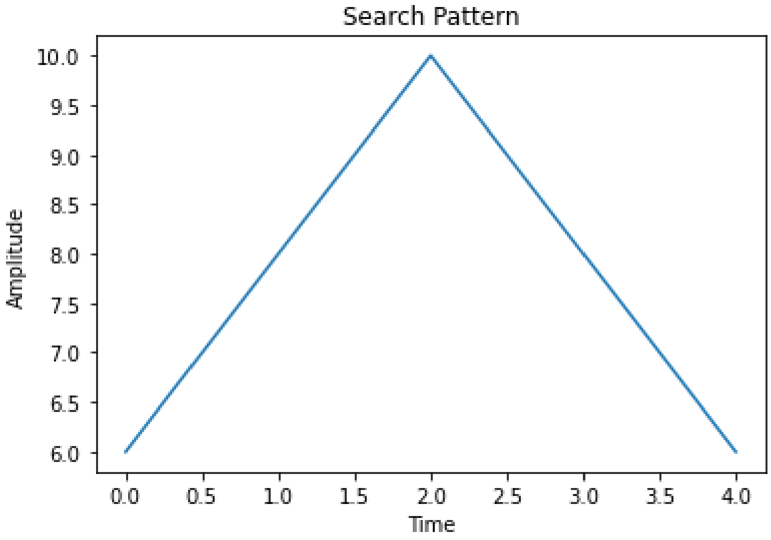


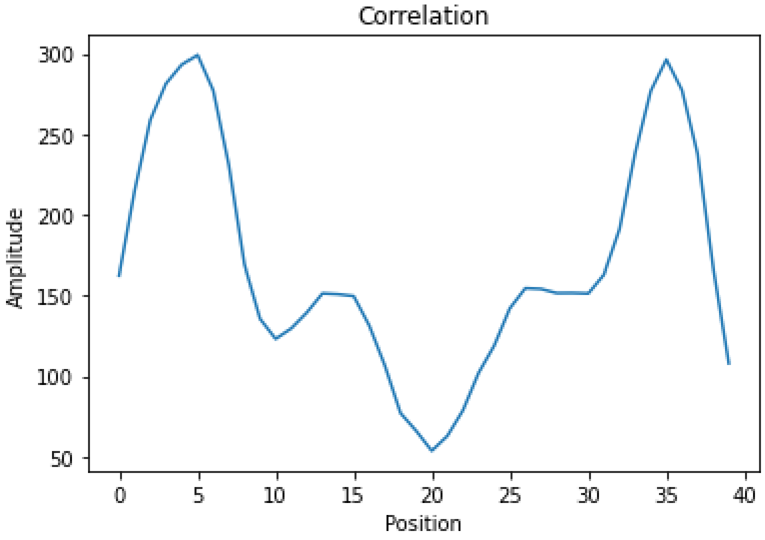
In [12]:

```
#Q4   NOISY SIGNAL - MULTIIPLE MATCH

import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

answer = []
Input = np.array([6, 8, 6, 6, 8, 10, 8, 6, 4 ,3, 2, 4, 4, 4, 4, 4, 4, 4, 1, 1,1, 3, 1,
2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 8, 10, 8, 6, 4, 4])
Input = Input + np.random.normal(0,0.1,len(Input))
search = np.array([6, 8, 10, 8, 6])
corr = signal.correlate(Input, search, mode = 'same')     #Correlation performed
length = len(corr)
Element = np.max(corr)
for i in range(0, length):
    if (corr[i] == Element):
        answer.append(i - 1)       #All matches will be stored in the output

#Showing where the patterns have been
print('Pattern found at starting locations: ', answer)

#Plotting the Input Signal
plt.plot(Input)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Input Signal')
plt.show()

#Plotting the Pattern to be searched
plt.plot(search)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Search Pattern')
plt.show()

#Plotting the Correlation Signal
plt.plot(corr)
plt.xlabel('Position')
plt.ylabel('Amplitude')
plt.title('Correlation')
plt.show()
```

```
Pattern found at starting locations:  [4]
```


Input Signal
Amplitude
Time


Search Pattern
Amplitude
Time


Correlation
Amplitude
Position