Question 3. On 'savgol-head.py', write your own Savitzky-Golay (SG) smoother (use the 5-point weighting, i.e. the 1st line of the table in the note). Apply it to the data in the same file.

In [1]:

```
import random
import scipy.signal as sg
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

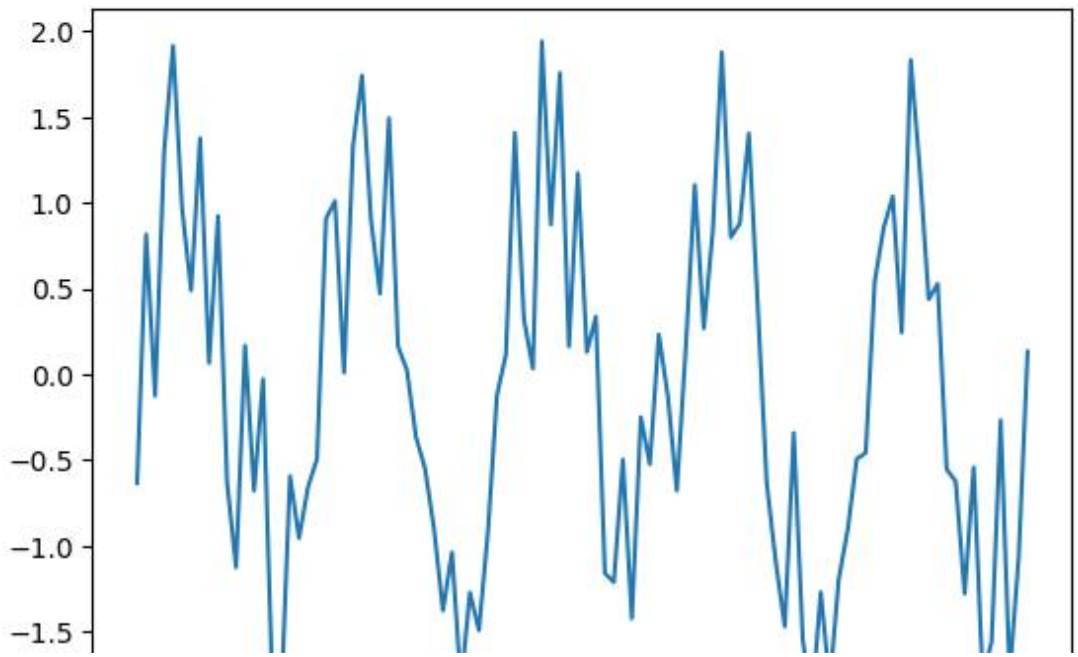
```
# this is a signal with random noise
x=np.arange(100)
y0=np.sin(10.0*np.pi*x/100)
y=np.sin(10.0*np.pi*x/100)
```

In [3]:

```
for i in x:
    y[i] += 2.0*(random.random()-0.5)
```

In [4]:

```
plt.plot(x,y); plt.show()
##############################3
```

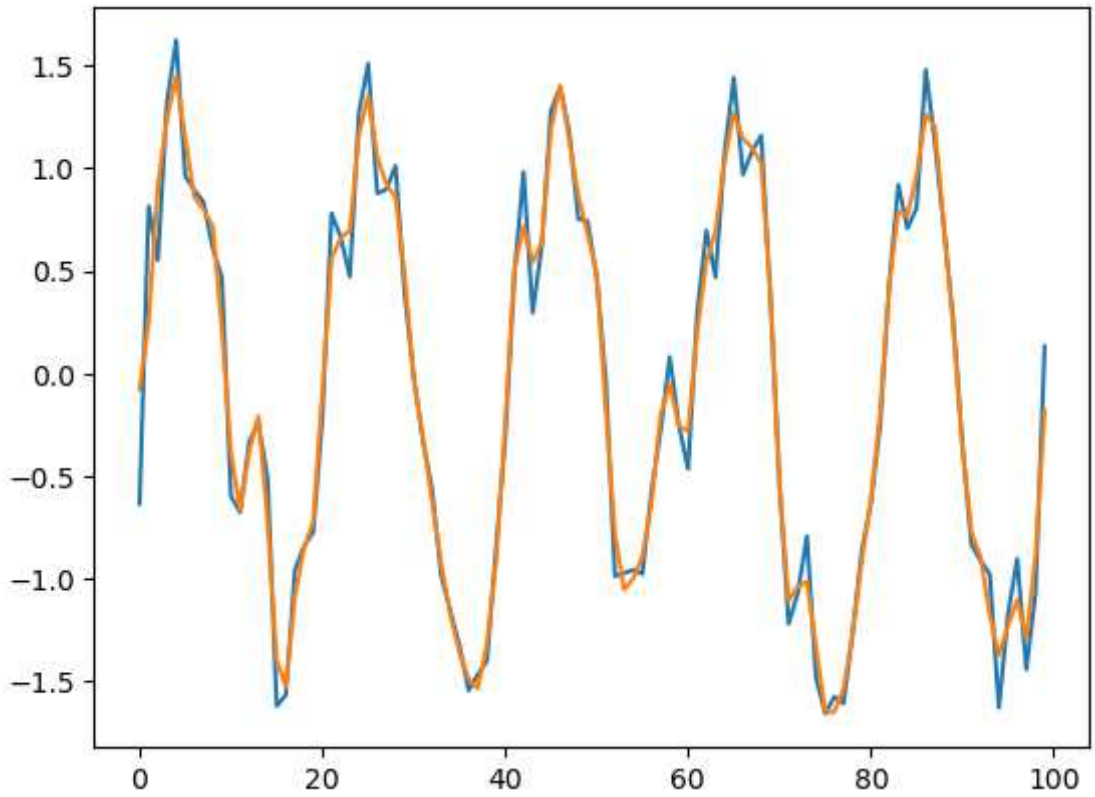


In [11]:

```
# Write your own sav-gol filter here and plot it along with the original signal
y2 = y*0
y2[0] =  0.486*y[0] +0.343*y[1]-0.086*y[2]
y2[1] = 0.343*y[0]+0.486*y[1] +0.343*y[2]-0.086*y[3]
y2[98] = -0.086*y[96] +0.343*y[97]+0.486*y[98] +0.343*y[99]
y2[99] = -0.086*y[97] +0.343*y[98]+0.486*y[99]
for i in range(2,100-2) :
        y2[i] = -0.086*y[i-2] +0.343*y[i-1]+0.486*y[i] +0.343*y[i+1]-0.086*y[i+2]
```

In [14]:

```
plt.plot(x,y)
plt.plot(x,y2); plt.show()
```



In [ ]:

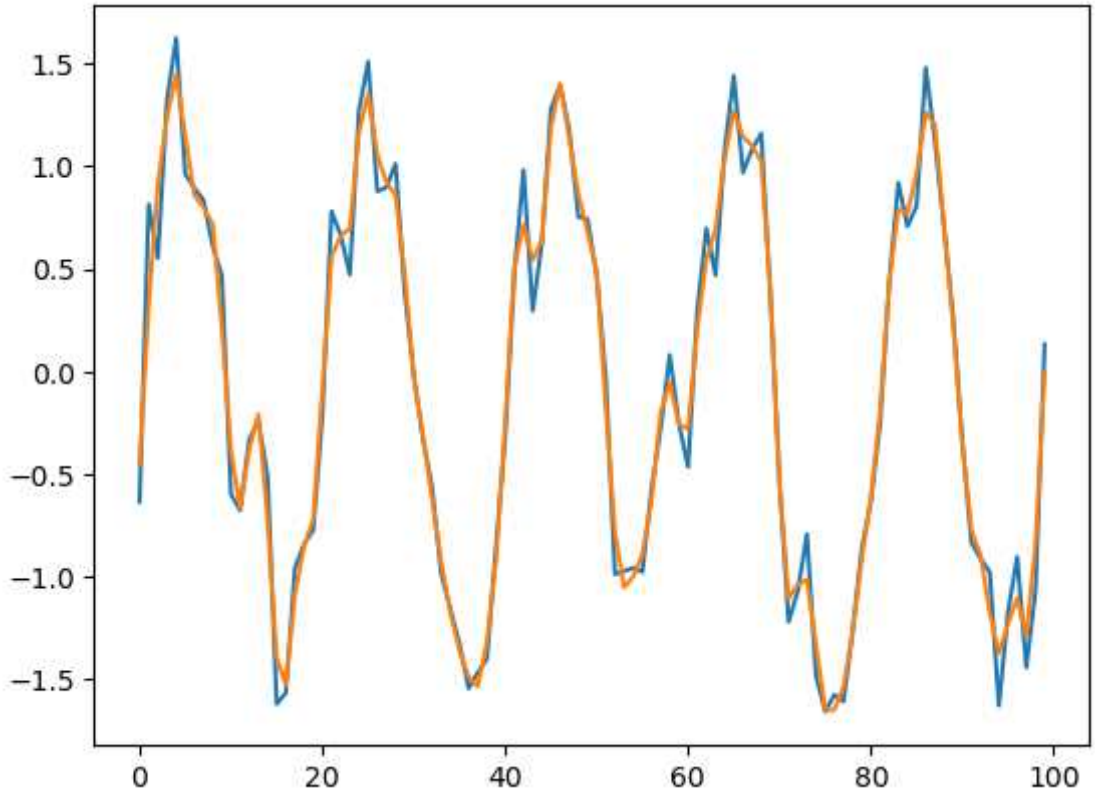
```
#built in functioon
```

In [15]:

```
from scipy.signal import savgol_filter
```

In [16]:

```
y3 = savgol_filter(y,5,2)
plt.plot(x,y)
plt.plot(x,y3); plt.show()
```



In [ ]:

```

```