Question 3. Find five energy levels from the ground for the one-dimensional double-well potential given by

$$V(x) = E_0(Cx^4 - x^2)$$

with

$$E_0 = 2 \times 10^{-4}, C = 0.045$$

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as integ
```

In [2]:

```
h=0.005
t=np.arange(0,12.0,h)
```

In [3]:

```
E0 = 2*10**-4
C = 0.045
# E0 = 1
# C = 1
```

In [4]:

```
def f(y,t,E):
    x,v=y
    U = E0*(C*t**4-t**2) # Set the potential
    return [v,2.0*(U-E)*x]
```

In [5]:

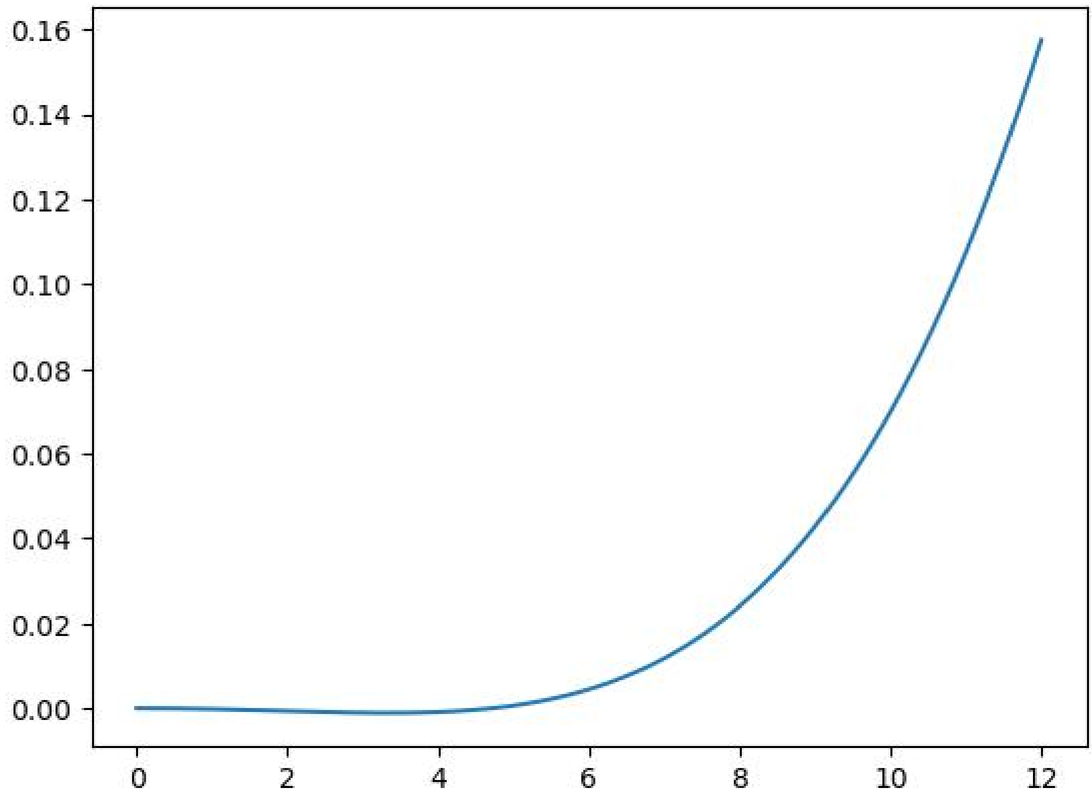
```
y0=[1.0,0.0]
```

In [6]:

```
#energy diagram
plt.plot(t,E0*(C*t**4-t**2))
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x183fb779370>]
```



In [7]:

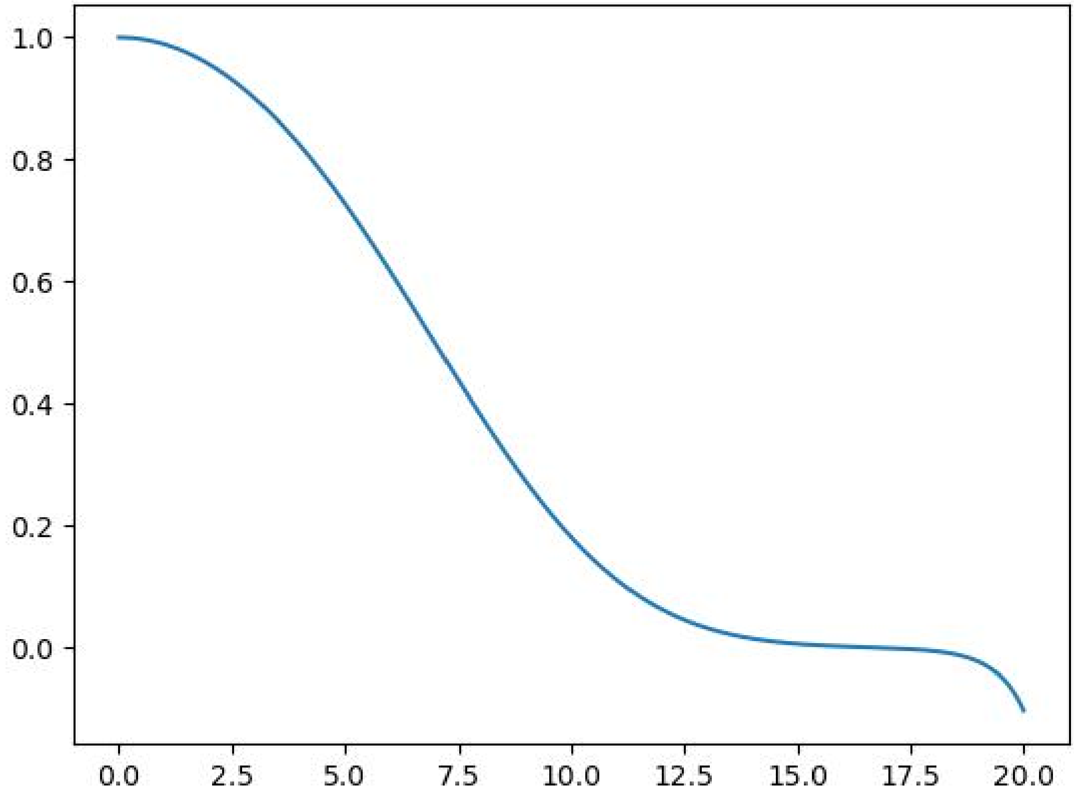
```
# find minimum of potential energy
np.min(E0*(C*t**4-t**2))
```

Out[7]:

```
-0.0011111099994443752
```

In [8]:

```
#example
E = 0.010964
y0 =[1.0,0.0]
te=np.arange(0,20.0,h)
sol = integ.odeint(f,y0,te,args=(E,))
plt.plot(te,sol[:,0]); plt.show()
```



By the potential which goes to the infinity, the wave function will go positive zero in infinity.Therefoer, if we find the energy which becomes negative, then the allowed energy will be between that energy and former energy.

In [9]:

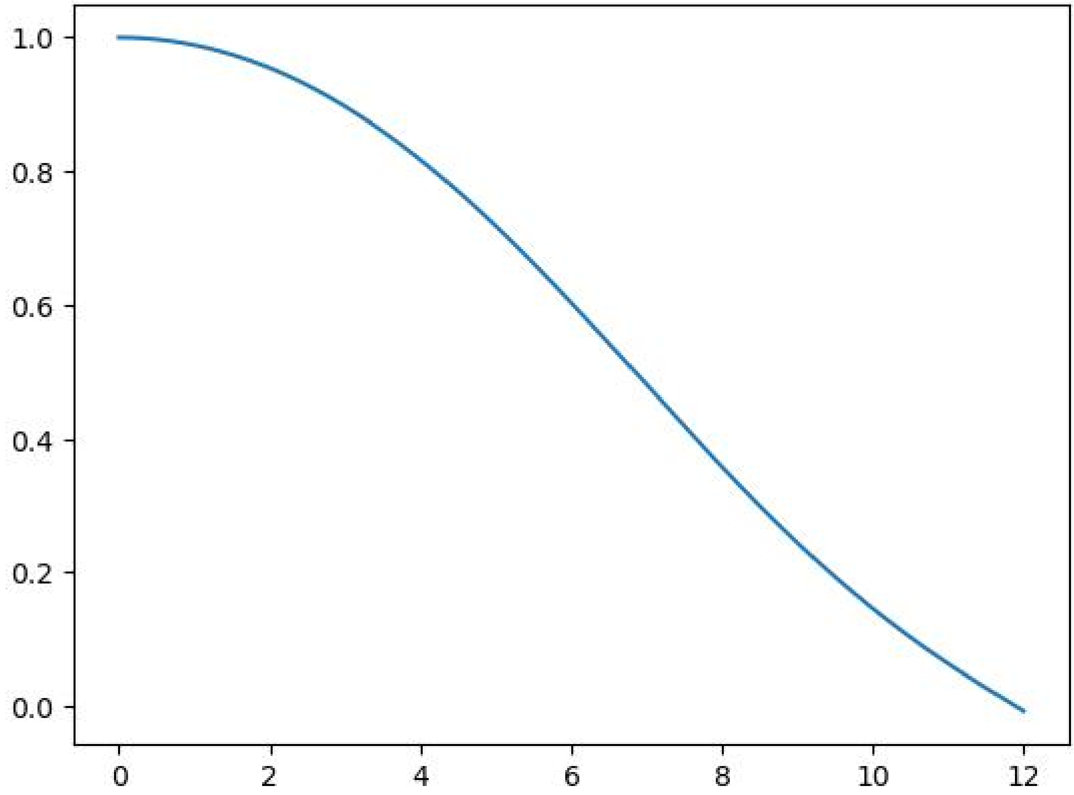
```
def findE(y0):
    h = 1e-4
    E = 0
    E_b = 0
    while True :
        sol = integ.odeint(f,y0,t,args=(E,))
        if sol[-1,0]<0:
            plt.plot(t,sol[:,0]); plt.show()
            break
        sol_before = sol
        E_b = E
        E = E + h
    return (E_b+E)/2
```

In [10]:

```
#ground state
y0=[1.0,0.0]
```

In [11]:

```
Eg = findE(y0)
```



In [12]:

```
Eg
```

Out[12]:

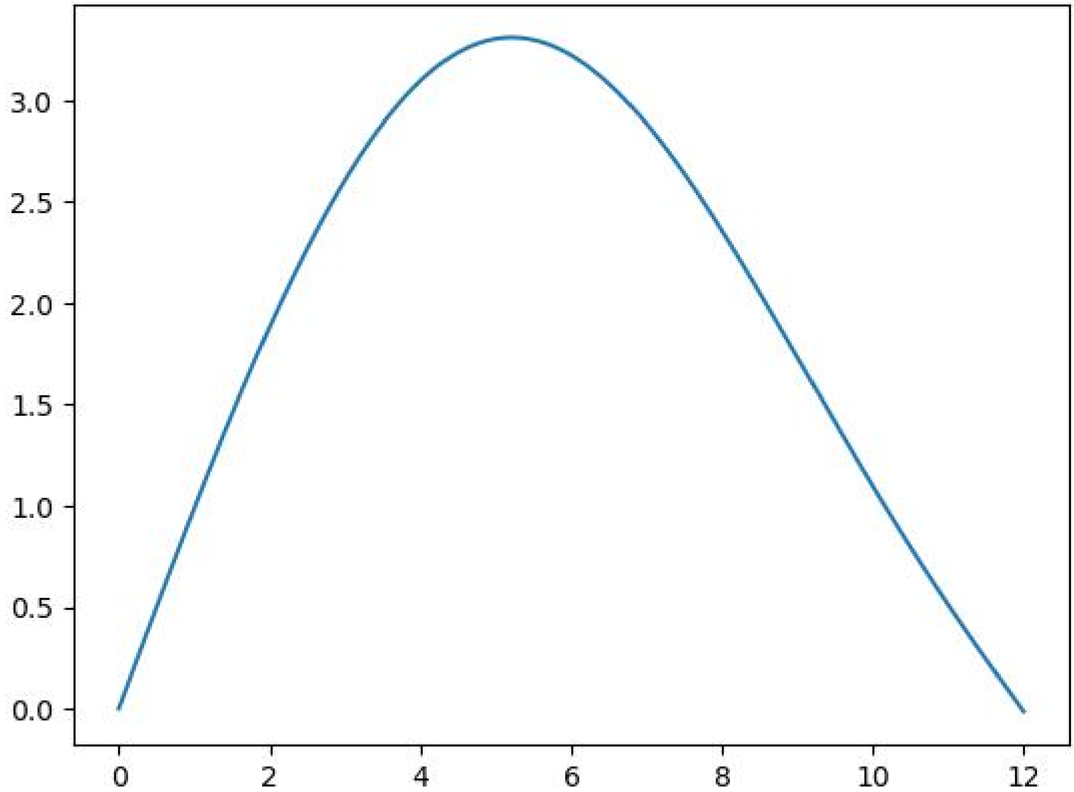
0.011349999999999987

In [13]:

```
#1st excite state
y0=[0.0,1.0]
```

In [14]:

```
E1 = findE(y0)
print(E1)
```



0.04485000000000026

In [15]:

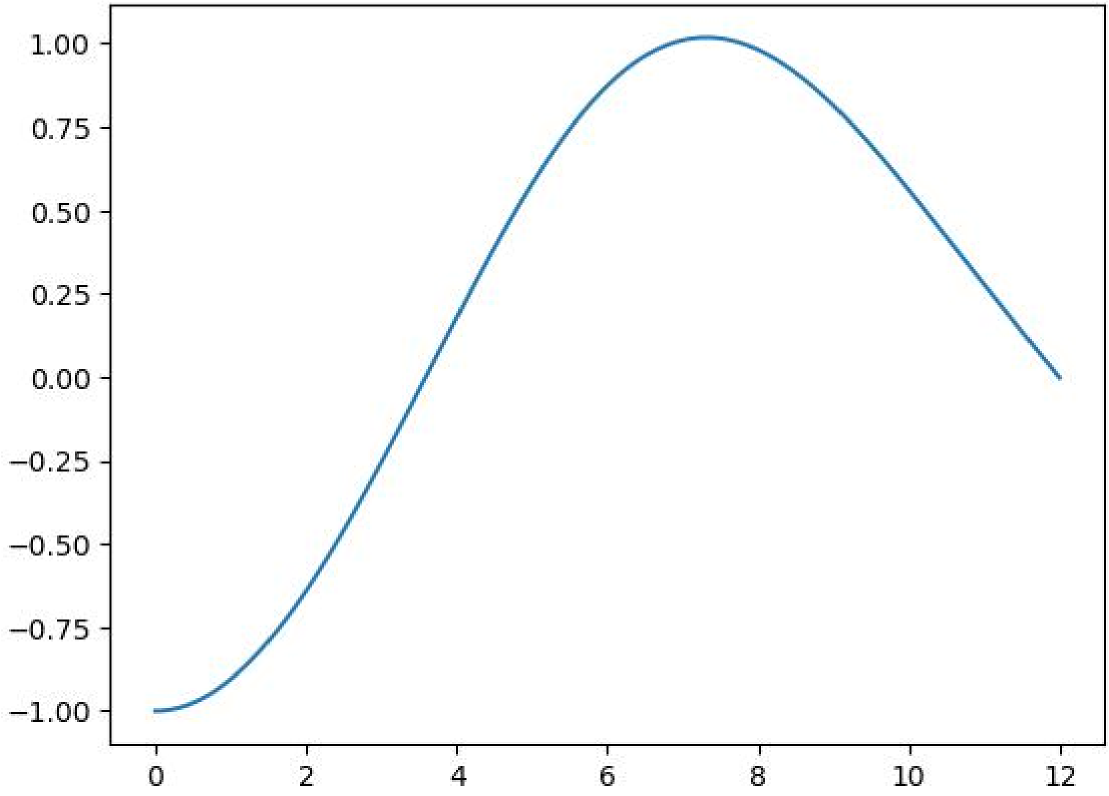
```
#2nd excite state
y0=[-1.0,0.0]
```

In [16]:

```
def findE2(y0):
    h = 1e-4
    E = 0.045 # after first excited state
    E_b = 0
    while True :
        sol = integ.odeint(f,y0,t,args=(E,))
        if sol[-1,0]<0:
            plt.plot(t,sol[:,0]); plt.show()
            break
        sol_before = sol
        E_b = E
        E = E + h
    return (E_b+E)/2
```

In [17]:

```
E2 = findE2(y0)
```



In [18]:

```
E2
```

Out[18]:

```
0.09555000000000172
```

In [19]:

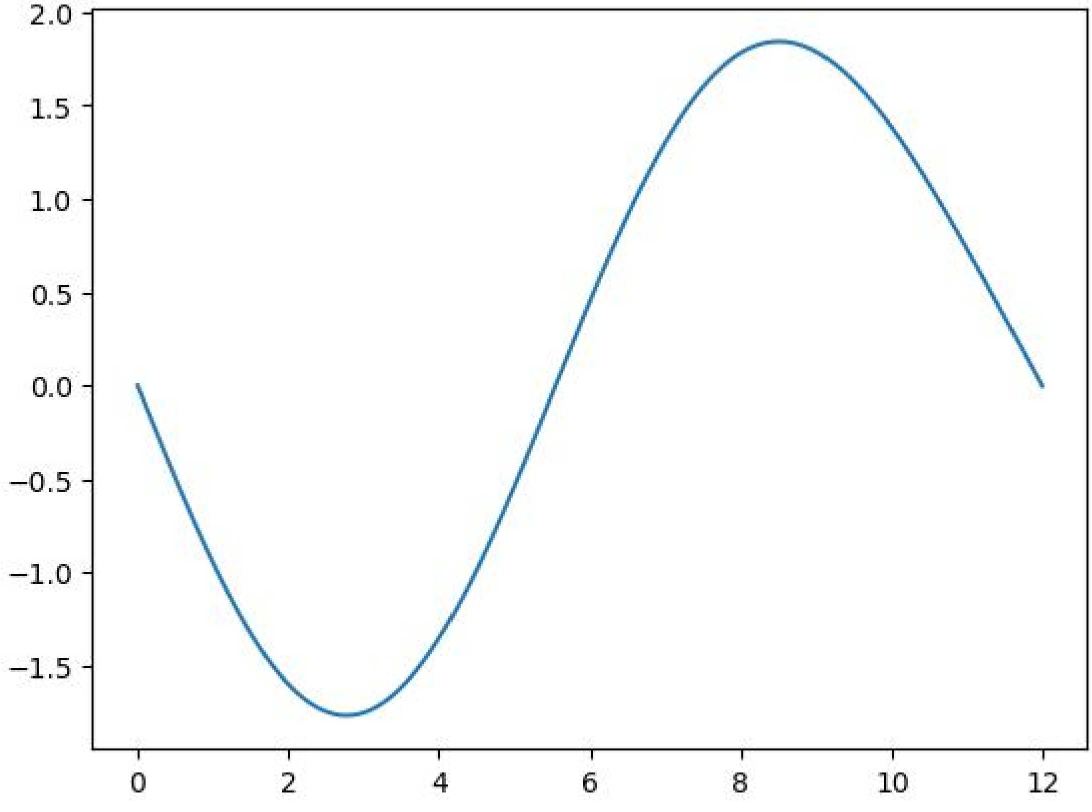
```
#3rd excite state
y0=[0.0,-1.0]
```

In [28]:

```
def findE3(y0):
    h = 1e-4
    E = 0.1 #after second excited state
    E_b = 0
    while True :
        sol = integ.odeint(f,y0,t,args=(E,))
        if sol[-1,0]<0:
            plt.plot(t,sol[:,0]); plt.show()
            break
        sol_before = sol
        E_b = E
        E = E + h
    return (E_b+E)/2
```

In [29]:

```
E3 = findE3(y0)
```



In [30]:

```
E3
```

Out[30]:

```
0.15994999999999687
```

In [33]:

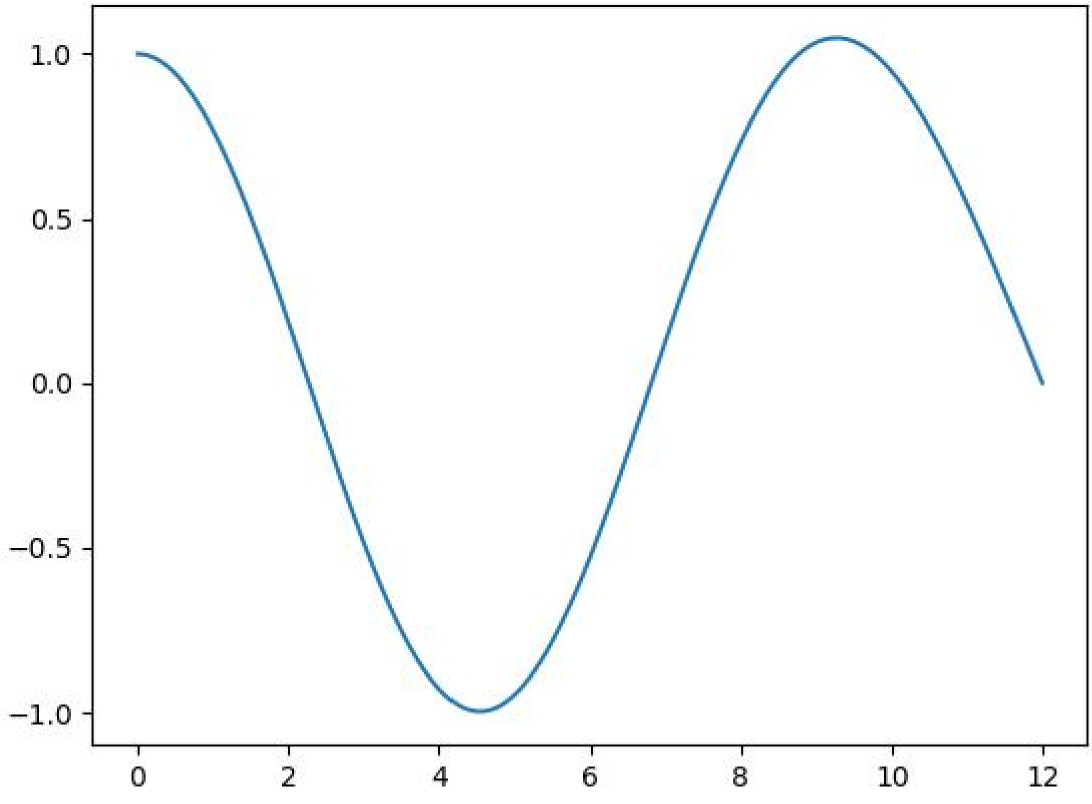
```
#4rd excite state
y0=[1.0,0.0]
```

In [34]:

```
def findE4(y0):
    h = 1e-4
    E = 0.16 # after third excited state
    E_b = 0
    while True :
        sol = integ.odeint(f,y0,t,args=(E,))
        if sol[-1,0]<0:
            plt.plot(t,sol[:,0]); plt.show()
            break
        sol_before = sol
        E_b = E
        E = E + h
    return (E_b+E)/2
```

In [35]:

```
E4 = findE4(y0)
```



In [36]:

```
E4
```

Out[36]:

```
0.23934999999999126
```