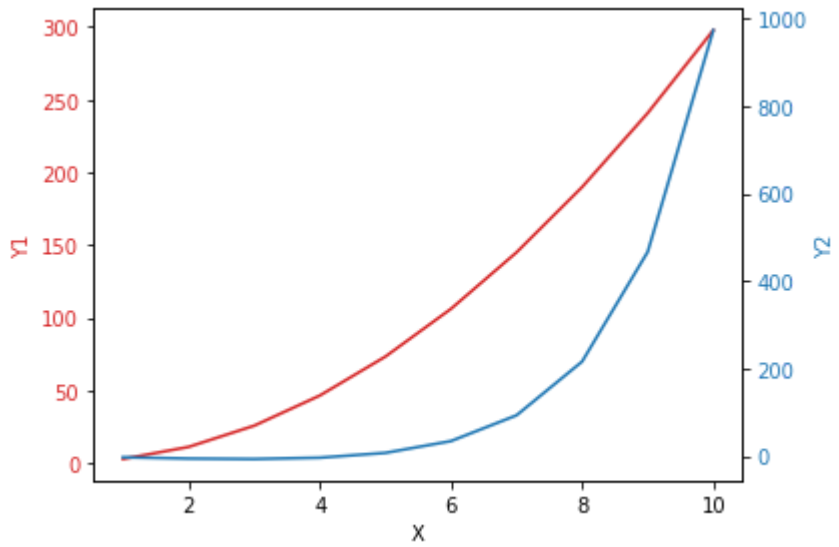
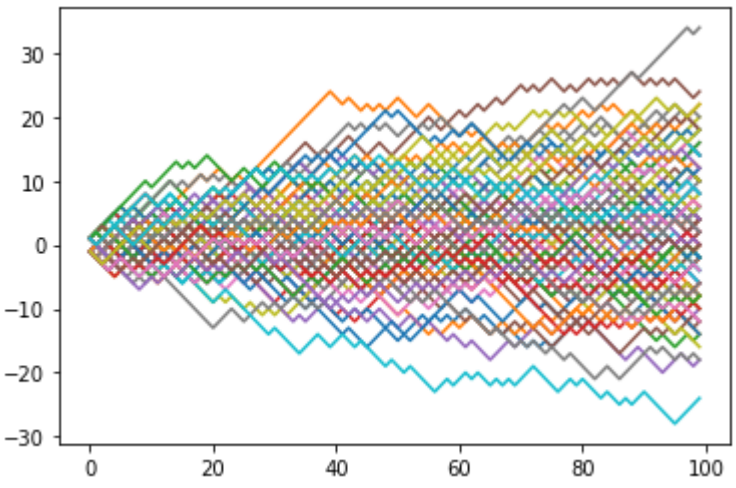
In [8]:
```
#By bhattarairaaju@gmail.com
#Task1: x=[1,...,10], y1=3x**2-log(x), y2=2**x -5x
# Create a dual axis plot. You might need to google on how to do it.
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(1,11)
y1=3*x**2-np.log(x)
y2=2**x -5*x

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('X')
ax1.set_ylabel('Y1', color=color)
ax1.plot(x, y1, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('Y2', color=color)  # we already handled the x-label with ax1
ax2.plot(x, y2, color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```



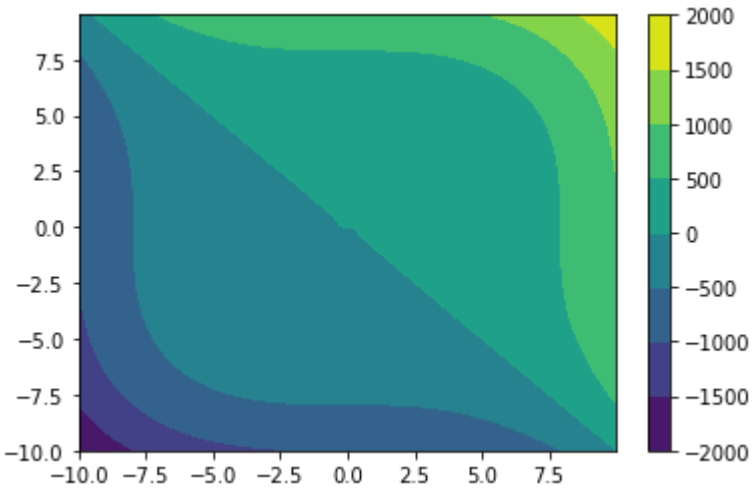
In [312]:
```
#Task2: Random Walk 1D
# 'A guy is drunk! He either goes one step forward or one step backward'.
# Where will he be in 100 steps?
# Create a loop to run this function 100 times.
# Then calculate the average final position.
avg=0
for a in range(100):
    import numpy as np
    import random
    def walk1D(x,N,d):
        X=[] # store position
        T=[] # store step
        y=[]
        for t in range(N):
            x=x+random.choice([-1,1])*d
            T.append(t)
            X.append(x)
        return T,X
    time,pos=walk1D(0,100,1)
    import matplotlib.pyplot as plt
    y=pos[99]
#    print(y)
    avg=avg+(y/2)
    plt.plot(time,pos)
print(avg)
```

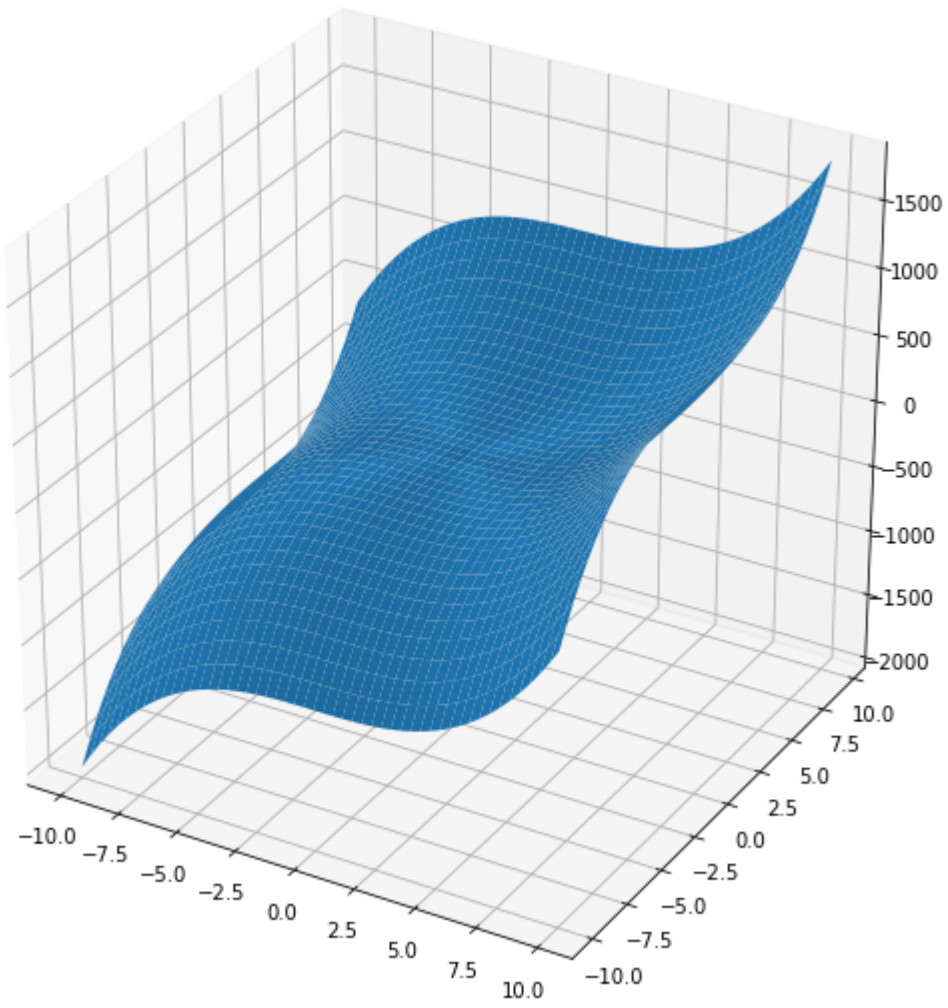
```
108.0
```



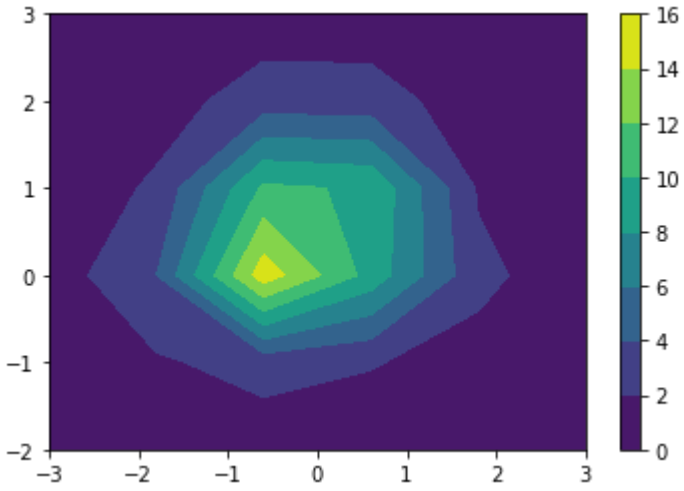
In [287]:
```
#Task3: Create a grid.
#Task4: Create counterplot and 3D plot.
x=np.arange(-10,10,0.05)
y=np.arange(-10,10,0.5)
grid=np.meshgrid(x,y)
print (len(x),len(y),len(grid),len(grid[0]))
xx,yy=np.meshgrid(x,y)
z=xx**3+yy**3
plt.contourf(x,y,z)
plt.colorbar()
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig=plt.figure(figsize=[10,10])
ax=fig.gca(projection='3d')
surf=ax.plot_surface(xx,yy,z)
```

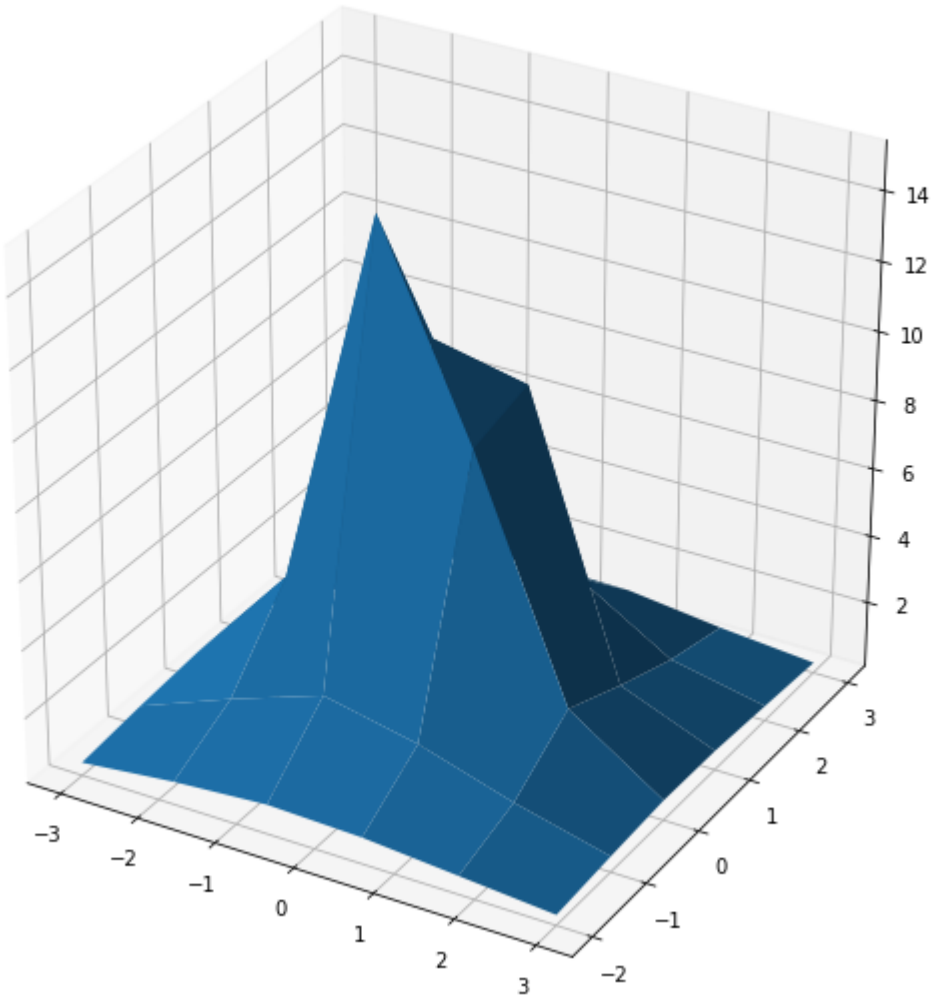
```
400 40 2 40
```




1500
1000
500
0
−500
−1000
−1500
−2000
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0

In [310]:
```
# Theoritically calcutate a formula to determine force at different points of the
#1,2, and 3 kg located at (1,0),(-1,0) and (0,1). G=1.
x=np.linspace(-3,3,6)
y=np.linspace(-2,3,6)
xx,yy=np.meshgrid(x,y)
m1=1
m2=2
m3=3
def f(i,j):
    F=(m1/((i-1)**2+(j-0)**2))+(m2/((i+1)**2+(j-0)**2))+(m3/((i-0)**2+(j-1)**2))
    return F
z=f(xx,yy)
plt.contourf(x,y,z)
plt.colorbar()
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig=plt.figure(figsize=[10,10])
ax=fig.gca(projection='3d')
surf=ax.plot_surface(xx,yy,z)
```





In [ ]: