ASSIGNMENT 7

PART 1

In [2]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:

```
#crseating dataset
def predictions(x,add_noise=False,mean =0,var = 0.25):
    if not add_noise:
        return x
    return x+np.random.normal(mean,var,x.shape)
x = np.linspace(-1,5,50)*1000  #(multiply by 1000)
```

In [4]:

```
data_norm = []
for i in range(len(x)):
    data_norm.append((x[i]-min(x))/(max(x)-min(x)))
data_norm = np.array(data_norm)
y = predictions(data_norm,True)
```

In [5]:

```
#mse
def mse(yt,yp):
    return (np.sum((yt-yp)**2)/len(yt))

def updates(yt,yp,x,lr,m,c):
    m = m- lr*((-x)*(np.sum(yt-yp)/len(yt))*2)
    c = c- lr*((-1)*(np.sum(yt-yp)/len(yt))*2)

    return m,c
```

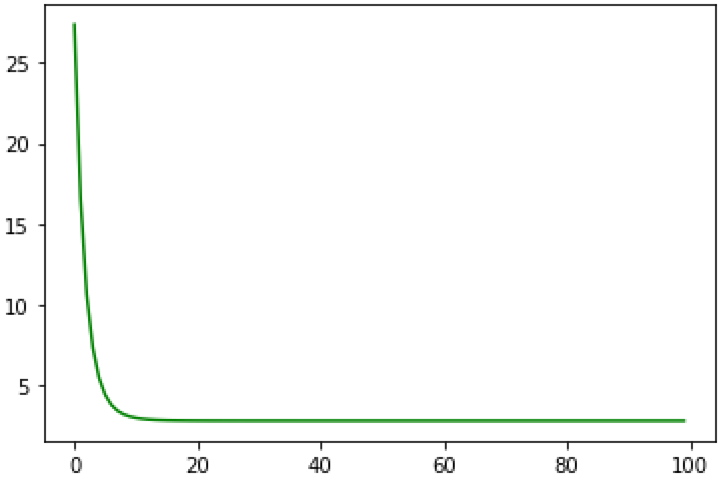
In [6]:

```
m,c = 10,0
lr = 0.1
total_loss = []
for i in range(100):
    yp = m*data_norm + c
    loss = mse(y,yp)
    total_loss.append(loss)
    m,c = updates(y,yp,data_norm,lr,m,c)

plt.show()
plt.plot(total_loss,color='green')
total_loss[-1], total_loss[0]
```

Out[6]:

```
(2.7800003734641763, 27.386306015734412)
```



In [ ]:

PART 2

In [7]:

```
data = pd.read_csv(r"C:\Users\VARNIKA\Desktop\Realestate.csv")
data
```

Out[7]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1
...	...	...	...	...	...	...	...	...
409	410	2013.000	13.7	4082.01500	0	24.94155	121.50381	15.4
410	411	2012.667	5.6	90.45606	9	24.97433	121.54310	50.0
411	412	2013.250	18.8	390.96960	7	24.97923	121.53986	40.6
412	413	2013.000	8.1	104.81010	5	24.96674	121.54067	52.5
413	414	2013.500	6.5	90.45606	9	24.97433	121.54310	63.9

414 rows × 8 columns

In [9]:

```
data = data.sample(frac=1)
x = data.drop('Y house price of unit area',axis=1)
y = np.array(data['Y house price of unit area'])
import math
ratio = 0.8
n_train = math.floor(ratio*x.shape[0])
```

```
n_test = math.ceil(1-ratio)*x.shape[0]
x_train = x[:n_train]
y_train = y[:n_train]
x_test = x[n_train:]
y_test = y[n_train:]
X =x.apply(lambda rec:(rec-rec.mean())/rec.std(),axis=0)
```

In [10]:

```
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

Out[10]:

```
((331, 7), (331,), (83, 7), (83,))
```

In [11]:

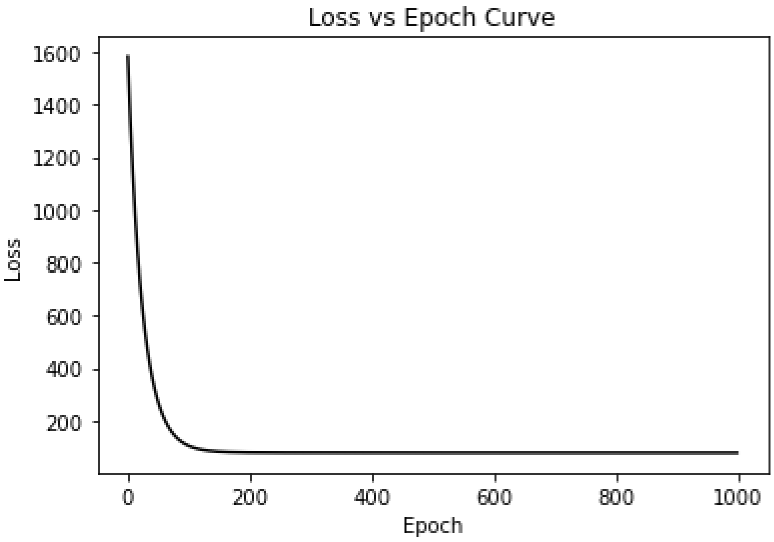
```
import random
def initialize(s):
    b=random.random()
    theta=np.random.rand(s)
    return b,theta
X.shape
```

Out[11]:

```
(414, 7)
```

In [12]:

```
b, theta = initialize(7)
l1 = []
l2 = []
l = 0.01
for i in range(1000):
    y_pred = b+np.dot(X,theta)
    e = np.mean((y-y_pred)**2)
    db = (np.mean(y-y_pred)*-2)
    dw = (np.dot((y-y_pred),X)*-2)/len(y)
    b = b-l*db
    theta = theta - l *dw
    l1.append(e)
    l2.append(i)
plt.plot(l2,l1,color='k')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Loss vs Epoch Curve')
plt.show()
```


Loss vs Epoch Curve
Loss
Epoch