In [16]:

```
import numpy as np
import matplotlib.pyplot as plt#question2
```

In [2]:

```
x=np.array([3,8,9,13,3,6,11,21,1,16])
```

In [3]:

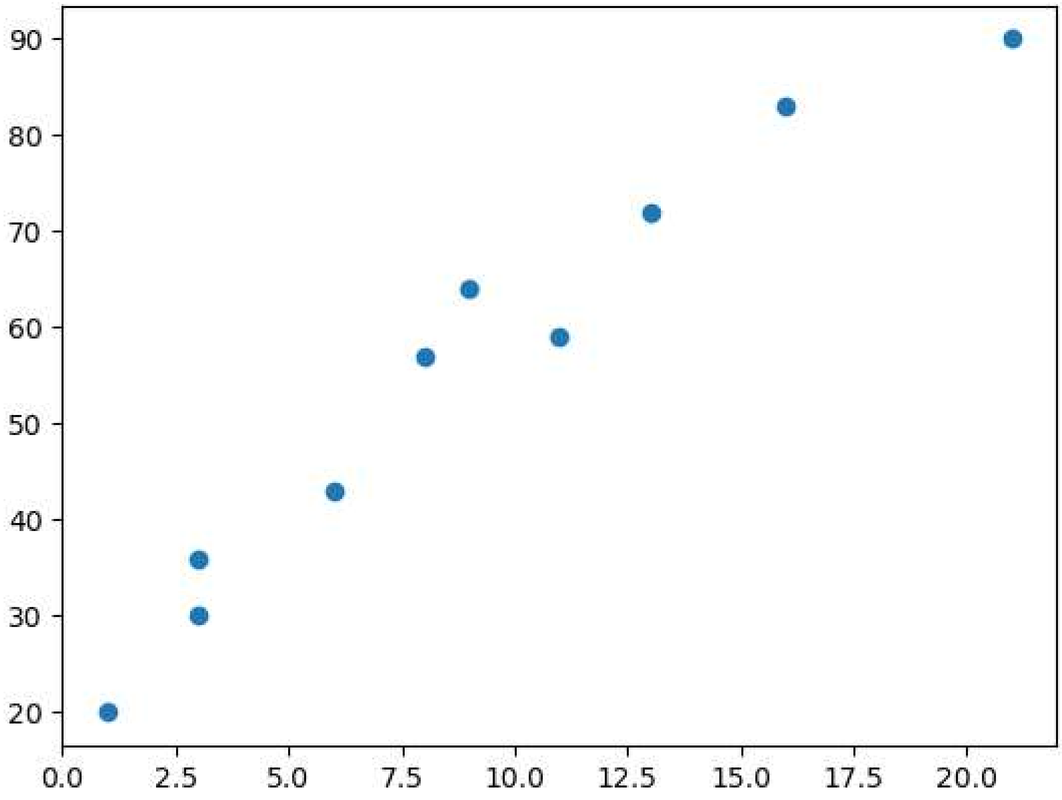
```
y=np.array([30,57,64,72,36,43,59,90,20,83])
```

In [4]:

```
plt.scatter(x,y)
```

Out[4]:

```
<matplotlib.collections.PathCollection at 0x138c82333d0>
```



In [5]:

```
meanx=x.mean()
```

In [6]:

```
meany=y.mean()
```

In [7]:

```
result1=((x-meanx)*(y-meany)).sum()
```

In [8]:

```
result2=((x-meanx)**2).sum()
```

In [9]:

```
slope=result1/result2
```

In [10]:

```
intercept=meany-slope*meanx
```

In [11]:

```
linex=np.array([i for i in range(1,21)])
```

In [12]:

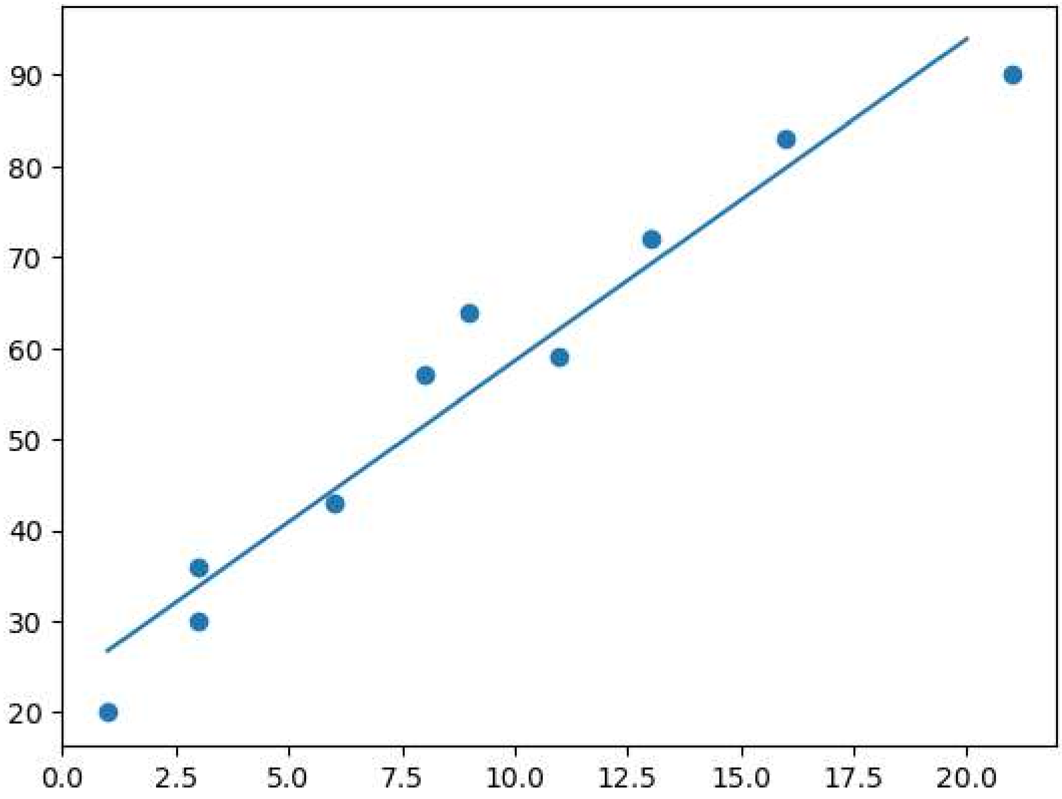
```
liney=np.array([slope*i+intercept for i in linex])
```

In [14]:

```
plt.scatter(x,y)
plt.plot(linex,liney)
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x138c8c61090>]
```



In [15]:

```
#question1
```

In [25]:

In [24]:

```
Original Data Shape: (10, 2)
Transformed Data Shape: (10, 1)
Transformed Data:
[[-0.82797019]
 [ 1.77758033]
 [-0.99219749]
 [-0.27421042]
 [-1.67580142]
 [-0.9129491 ]
 [ 0.09910944]
 [ 1.14457216]
 [ 0.43804614]
 [ 1.22382056]]
```

In [29]:

```
import numpy as np
from sklearn.decomposition import PCA


data = np.array([[2.5, 2.4],
                 [0.5, 0.7],
                 [2.2, 2.9],
                 [1.9, 2.2],
                 [3.1, 3.0],
                 [2.3, 2.7],
                 [2.0, 1.6],
                 [1.0, 1.1],
                 [1.5, 1.6],
                 [1.1, 0.9]])

n_components = 1
pca = PCA(n_components=n_components)

transformed_data = pca.fit_transform(data)

print("Original Data Shape:", data.shape)
print("Transformed Data Shape:", transformed_data.shape)
print("Transformed Data:")
print(transformed_data)
```

```
Original Data Shape: (10, 2)
Transformed Data Shape: (10, 1)
Transformed Data:
[[-0.82797019]
 [ 1.77758033]
 [-0.99219749]
 [-0.27421042]
 [-1.67580142]
 [-0.9129491 ]
 [ 0.09910944]
 [ 1.14457216]
 [ 0.43804614]
 [ 1.22382056]]
```

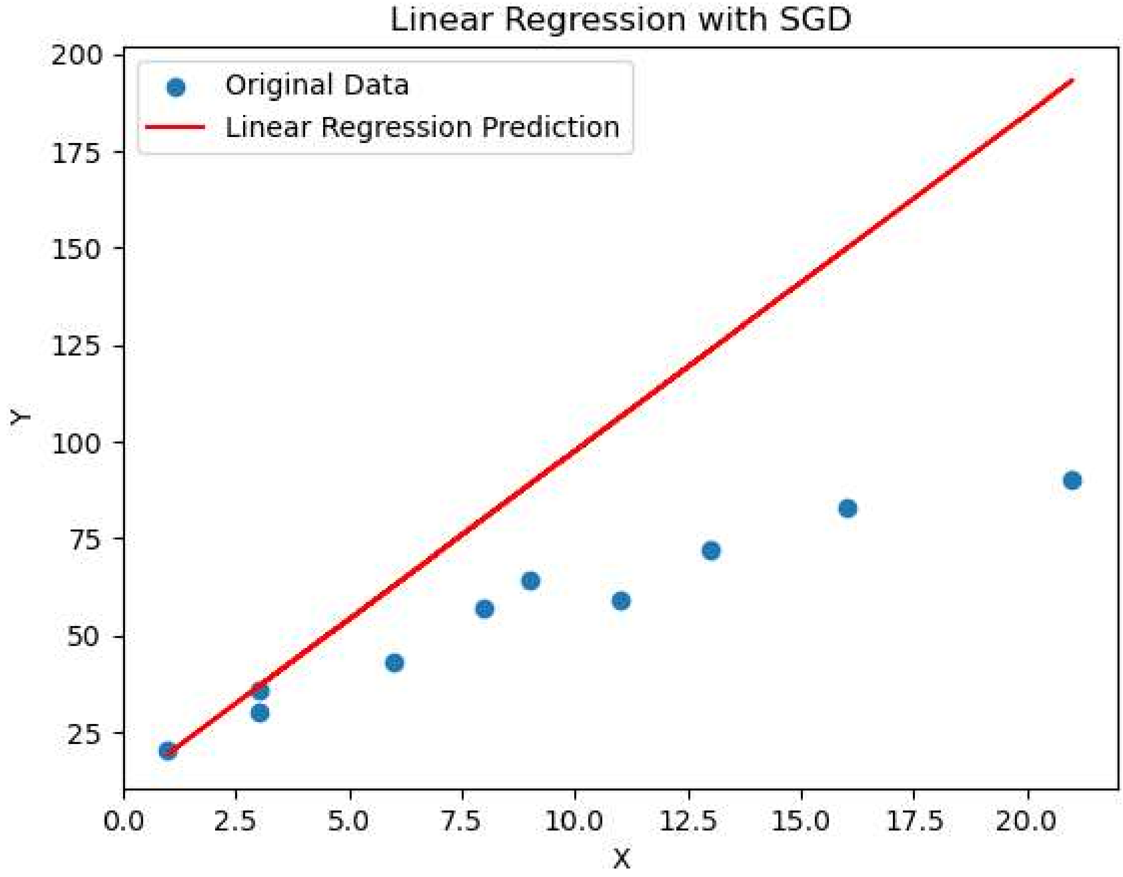
In [30]:

```
#question 3
```

In [31]:

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


X = np.array([3, 8, 9, 13, 3, 6, 11, 21, 1, 16])
Y = np.array([30, 57, 64, 72, 36, 43, 59, 90, 20, 83])

learning_rate = 0.01
epochs = 100

slope = 0
intercept = 0


for epoch in range(epochs):
    for i in range(len(X)):
        y_pred = slope * X[i] + intercept
        error = y_pred - Y[i]


        slope -= learning_rate * (error * X[i])
        intercept -= learning_rate * error


predictions = slope * X + intercept
```

```
plt.scatter(X, Y, label='Original Data')
plt.plot(X, predictions, color='red', label='Linear Regression Prediction')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear Regression with SGD')
plt.legend()
plt.show()
```



```
In [ ]:
```