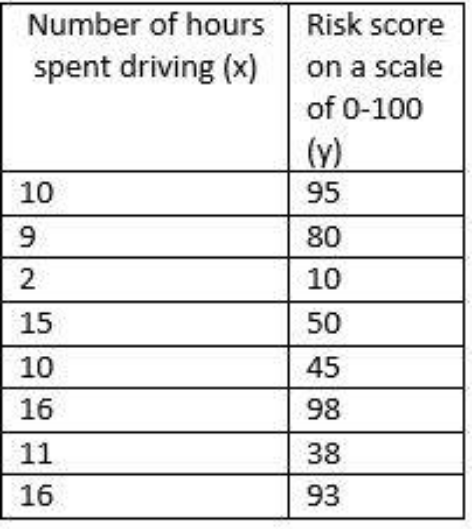
NAME: Kshitija Nalawade

ROLL NO: 4202041

BATCH: BB2

**Problem Statement:** The following table shows the results of a recently conducted study on the correlation of the number of hours spent driving with the risk of developing acute backache. Find the equation of the best fit line for this data.

****

**CODE:**

In [1]:

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**import** **pandas** **as** **pd**

In [2]:

*#input data*

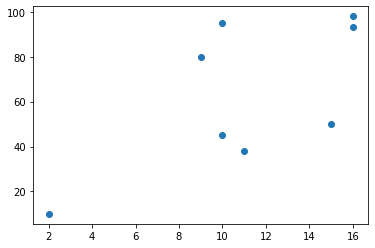
x = np.array([10,9,2,15,10,16,11,16])

y=np.array([95,80,10,50,45,98,38,93])

plt.scatter(x,y)

Out[2]:

<matplotlib.collections.PathCollection at 0x27845f93ee0>



In [3]:

**def** estimate\_coefficient(x,y):

*#Number of oberservations*

N =np.size(x)

*#calculate mean of x and y*

x\_mean,y\_mean=np.mean(x),np.mean(y);

*# calculating cross-deviation and deviation about x*

ss\_xy=np.sum(y\*x)-N\*x\_mean\*y\_mean;

ss\_xx=np.sum(x\*x)-N\*x\_mean\*x\_mean;

*#calculating regression coefficients*

b1=ss\_xy/ss\_xx;

b0=y\_mean-b1\*x\_mean;

**return** (b0,b1);

In [4]:

**def** plot\_regression\_line(x,y,b):

*#plotting actual points as scatter points*

plt.scatter(x,y, color="m", marker="o",s=30)

*#predicted response vector*

y\_pred=b[0]+b[1]\*x;

plt.plot(x,y\_pred, color="g");

*#putting labels*

plt.xlabel("X")

plt.ylabel("Y")

*#function to show plot*

plt.show()

In [5]:

b=estimate\_coefficient(x,y)

print("Estimated coefficients:**\n** b0 = **{}** **\n** b1 = **{}**".format(b[0],b[1]))

Estimated coefficients:

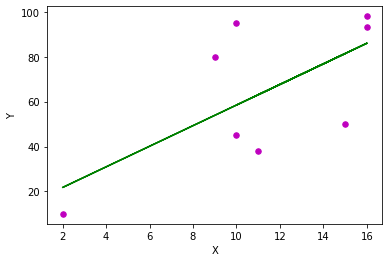
b0 = 12.584627964022893

b1 = 4.58789860997547

In [6]:

*# plotting regression line*

plot\_regression\_line(x,y,b)



In [7]:

*#END OF PROGRAM*

*#Below code for checking sklearns model*

**from** **sklearn.linear\_model** **import** LinearRegression

In [8]:

model=LinearRegression().fit(np.reshape(x,(-1,2)),np.reshape(y,(-1,2)))

model.coef\_

Out[8]:

array([[ 4.5314367 , -7.99445235],

[ 5.03755334, 1.83015647]])

In [9]:

model.intercept\_

Out[9]:

array([121.53798009, 13.06799431])

In [10]:

model.score(np.reshape(x,(-1,2)),np.reshape(y,(-1,2)))

Out[10]:

0.9728057213483525