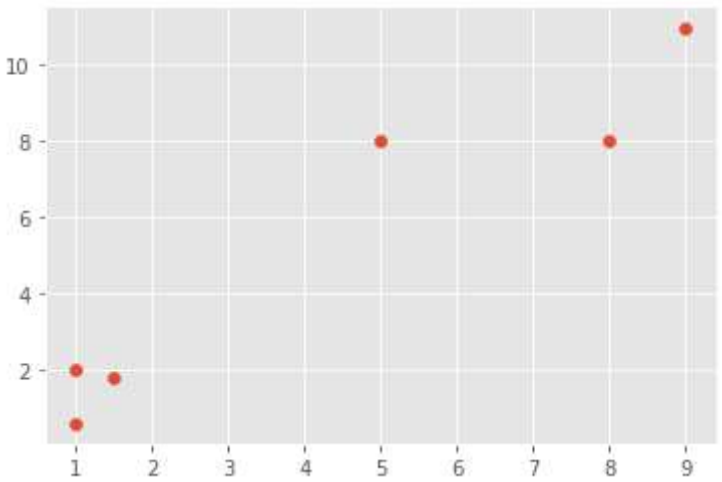
In [5]:
```
## importing Numpy, Matplotlib and sklearn libraries
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import style
style.use("ggplot")
from sklearn.cluster import KMeans

# Plotting and visualizing our data before feeding it into the Machine Learnin
g Algorithm
x = [1,5,1.5,8,1,9]
y = [2,8,1.8,8,0.6,11]
plt.scatter(x,y)
plt.show()
```

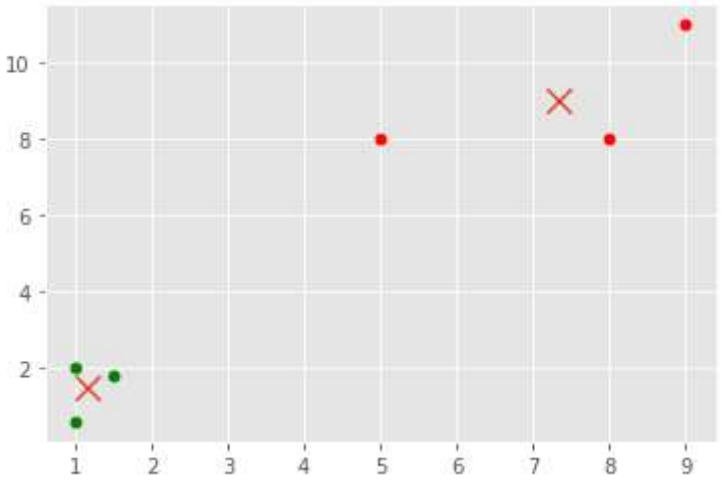


In [8]:
```
## Converting our data into a NumPy array
x = np.array([[1,2],[5,8],[1.5,1.8],[8,8],[1,0.6],[9,11]])

## We initialize K-means algorithm with the required parameter and we use .fit
() to fit the data
kmeans = KMeans(n_clusters=2)
kmeans.fit(x)

## Getting the values of centroids and labels based on the fitment

centroids = kmeans.cluster_centers_
labels = kmeans.labels_

print(centroids) ## (x,y) values for the 2 centroids
print(labels) ## for each (x,y) value this label says which centroid it will b
eling to
```

```
[[1.16666667 1.46666667]
 [7.33333333 9.        ]]
[0 1 0 1 0 1]
```

In [11]:
```
## Plotting and visualizing output

colors = ["g.","r.","c.","y."]

for i in range(len(x)):
    print("coordinate:",x[i], "label:", labels[i])
    plt.plot(x[i][0], x[i][1], colors[labels[i]], markersize = 10)

plt.scatter(centroids[:, 0],centroids[:, 1], marker = "x", s=150, linewidths =
5, zorder = 10)

plt.show()
```

```
coordinate: [1. 2.] label: 0
coordinate: [5. 8.] label: 1
coordinate: [1.5 1.8] label: 0
coordinate: [8. 8.] label: 1
coordinate: [1.  0.6] label: 0
coordinate: [ 9. 11.] label: 1
```



In [ ]: