```
#importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

```
#creating data frame
exams_data = pd.read_csv('/content/drive/MyDrive/Data/exams.csv')
```

```
#for printing size of imported dataset
exams_data.shape
```

```
(1000, 8)
```

```
#printing first five values of dataset
exams_data.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	rea s
0	male	group A	high school	standard	completed	67	
1	female	group D	some high school	free/reduced	none	40	
2	male	group E	some college	free/reduced	none	59	
3	male	group B	high school	standard	none	77	

```
#printing information of dataset
exams_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

```
#checking whether the dataset contains null value or not
exams_data.isnull().sum()
```

```
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

```
#choosing reading and writing scores columns from dataset
X = exams_data.iloc[:,[6,7]].values
print(X)
```

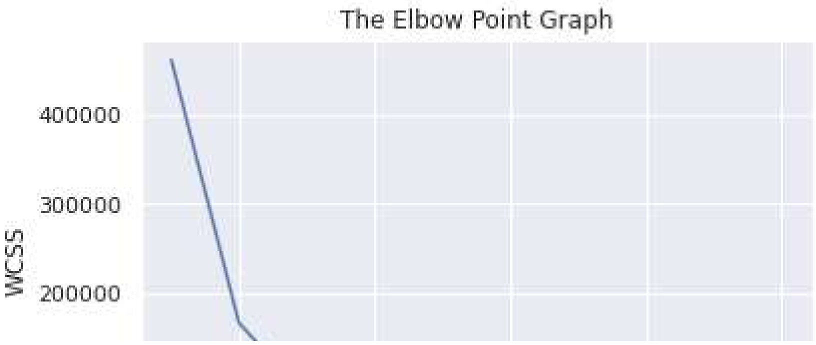
```
[[67 63]
 [59 55]
 [60 50]
 ...
 [35 41]
 [74 82]
 [60 62]]
```

```
#Within Clusters : Sum of Squares
#For diff number of clusters like 2,3, .... 10.

#Creating an empty list
wcss = []

for i in range(1,11):
  kmeans = KMeans(n_clusters=i, init='k-means++', random_state = 42)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_)
```

```
#plotting elbow graph
sns.set()
plt.plot(range(1,11), wcss)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```



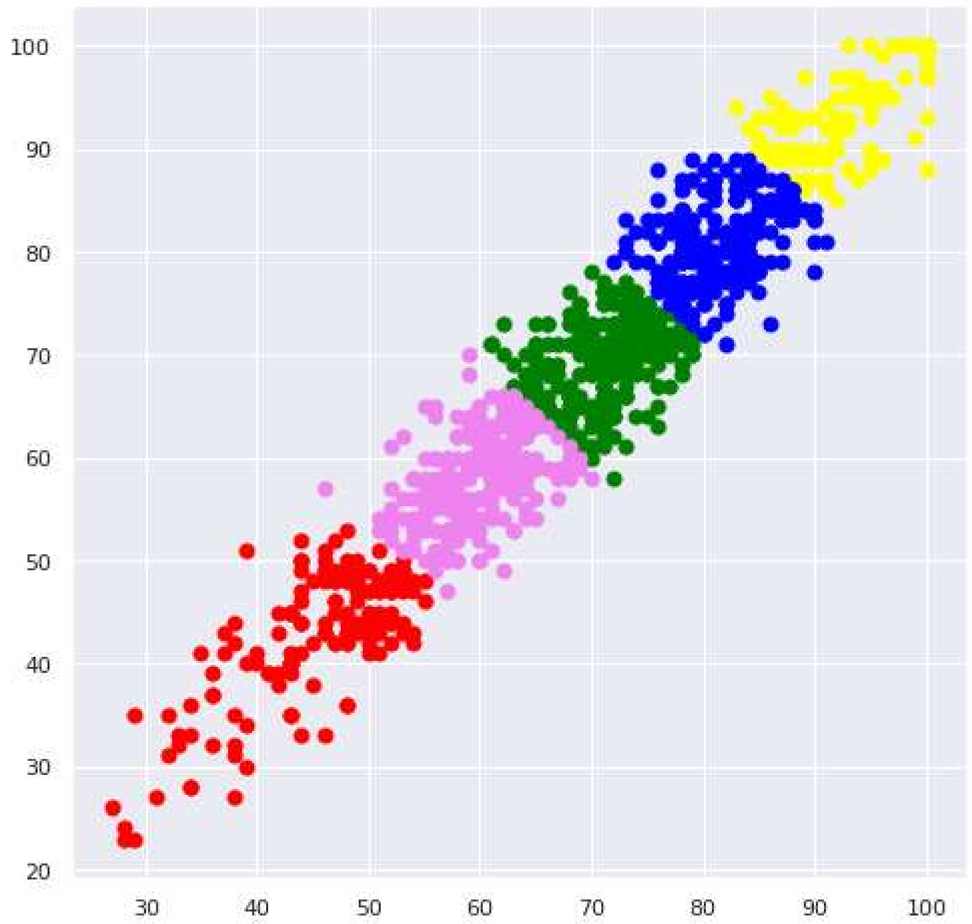
```
#Training the KMeans cluster model
kmeans = KMeans(n_clusters=5, init='k-means++', random_state = 0)

#Return a lable for each datapoint based on their cluster
Y = kmeans.fit_predict(X)
print(Y)
```

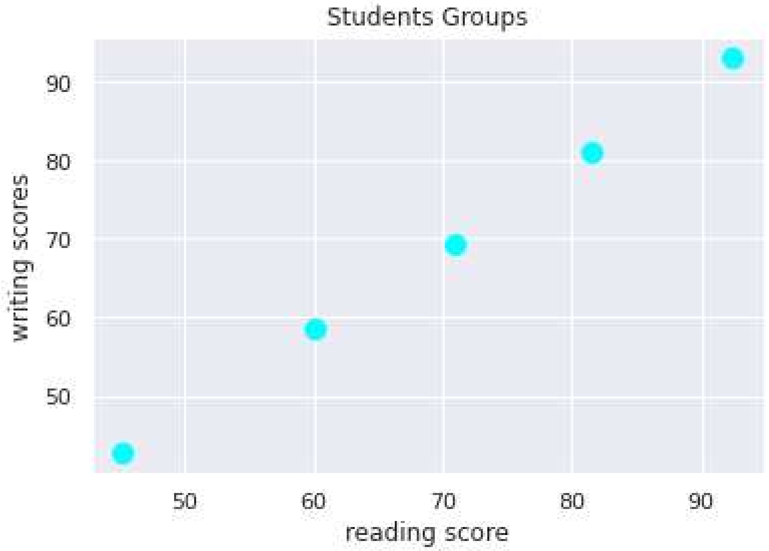
```
    [0 4 4 0 0 3 4 3 4 1 3 2 3 0 3 4 1 4 1 1 3 4 0 4 1 1 4 4 2 4 3 0 1 0 2 0 3
     4 0 3 0 0 3 0 2 1 4 3 0 4 0 4 3 1 4 4 3 3 4 1 0 4 4 3 2 0 4 1 4 4 0 1 0 3
     0 0 4 3 0 1 4 1 4 0 1 0 0 2 4 0 1 0 3 2 0 0 2 1 1 4 4 3 3 0 4 3 0 1 0 1 3
     1 3 1 0 3 4 4 4 2 0 2 4 1 3 4 0 1 3 2 3 0 4 4 4 0 3 0 1 0 4 3 3 4 0 4 4 1
     2 1 3 4 3 0 4 1 4 3 2 3 2 4 2 0 0 2 4 0 2 3 4 0 0 4 0 4 4 4 2 3 4 1 2 3 3
     2 0 4 4 4 4 1 4 2 0 1 1 4 0 4 4 1 0 0 3 0 3 3 3 4 0 3 0 0 3 4 0 3 4 4 3 0
     1 3 2 0 1 0 3 3 3 0 1 0 3 2 0 2 4 0 3 3 4 4 1 0 1 0 4 3 0 0 4 0 4 4 3 0 0
     2 0 1 4 4 3 0 0 3 4 0 0 0 0 2 2 3 0 3 1 0 3 2 0 1 1 3 3 1 0 4 0 3 0 0 4 3
     1 4 4 1 1 1 4 0 4 4 4 0 3 3 0 4 1 3 0 2 0 3 2 3 4 3 4 3 3 4 3 4 4 4 0 3 1
     3 1 0 4 1 0 3 4 1 4 0 3 4 3 0 4 2 4 1 3 0 0 4 0 4 0 1 3 1 4 3 0 0 4 0 4 0
     0 0 0 4 3 0 2 1 2 2 1 2 3 3 3 3 4 4 3 3 4 4 1 4 4 3 2 3 3 0 0 0 0 4 3 4 2
     0 1 0 0 0 4 4 4 0 4 0 4 1 0 0 1 0 1 3 0 0 1 3 3 0 0 2 4 3 0 1 2 0 4 2 0 1
     2 0 0 0 4 0 4 4 3 1 0 4 2 3 4 4 4 1 1 0 4 4 3 0 0 4 2 3 0 1 0 0 3 4 0 1 3
     3 3 3 2 2 4 4 3 0 0 0 1 3 4 4 1 0 2 3 4 2 4 3 1 2 0 4 4 0 0 3 0 3 3 4 1 1
     3 2 3 3 0 3 1 0 0 2 4 0 2 0 4 0 3 2 4 0 0 0 0 3 3 4 3 0 4 4 1 4 2 0 2 4 0
     1 1 0 3 1 4 4 0 4 2 3 4 3 1 0 4 2 1 3 4 2 4 4 0 1 3 0 0 3 3 4 0 2 2 0 3 2
     0 2 4 4 2 4 1 2 4 0 1 0 2 1 3 0 1 0 4 3 0 3 0 1 2 0 4 0 4 1 3 3 3 4 2 3 3
     4 0 4 4 4 2 2 4 0 3 0 3 0 0 4 1 4 3 0 2 1 0 0 2 1 4 3 0 0 0 1 2 0 2 3 3 0
     3 4 1 1 1 3 1 0 4 0 4 0 0 4 1 1 3 0 0 4 0 0 0 1 2 3 3 4 2 4 4 0 3 0 1 1 0
     0 0 4 2 3 3 0 4 3 1 4 0 4 2 1 4 1 1 2 4 4 4 1 3 0 4 1 3 0 2 0 4 2 2 3 0 0
     4 0 3 2 3 0 1 2 2 0 0 2 4 4 2 3 2 4 4 3 0 3 4 2 0 1 0 3 1 4 0 3 3 3 0 1 1
     0 1 1 2 3 2 4 3 4 0 3 4 4 3 0 0 0 3 1 2 4 4 4 3 4 2 1 0 1 0 0 3 4 1 4 4 4
     2 4 2 0 3 0 3 4 3 0 0 2 2 3 3 1 1 1 1 3 3 3 0 0 0 3 1 0 2 3 0 0 4 0 2 0 3
     3 4 0 2 2 4 2 2 3 2 3 3 0 3 0 4 3 4 4 4 1 3 4 0 4 4 0 4 1 0 3 4 3 3 1 0 0
     4 3 4 0 1 1 3 3 4 0 3 1 1 0 0 4 0 3 3 0 1 0 4 4 4 0 4 0 0 0 0 3 4 0 4 4 0
     4 0 0 4 3 3 1 3 0 4 0 1 1 1 4 3 3 3 0 0 4 0 4 1 4 2 0 4 4 4 3 0 4 4 3 4 3
     1 0 3 2 0 2 1 4 2 1 0 0 4 0 1 4 1 0 1 1 0 1 0 0 3 3 3 4 4 2 4 4 3 0 2 1 3
     4]
```

```
#plotting graphs for all clusters
plt.figure(figsize=(8,8))
plt.scatter(X[Y==0,0], X[Y==0,1], s=50, c='green', label='Cluster 1')
plt.scatter(X[Y==1,0], X[Y==1,1], s=50, c='red', label='Cluster 2')
plt.scatter(X[Y==2,0], X[Y==2,1], s=50, c='yellow', label='Cluster 3')
plt.scatter(X[Y==3,0], X[Y==3,1], s=50, c='blue', label='Cluster 4')
plt.scatter(X[Y==4,0], X[Y==4,1], s=50, c='violet', label='Cluster 5')
```

```
<matplotlib.collections.PathCollection at 0x7fdd69cd1090>
```

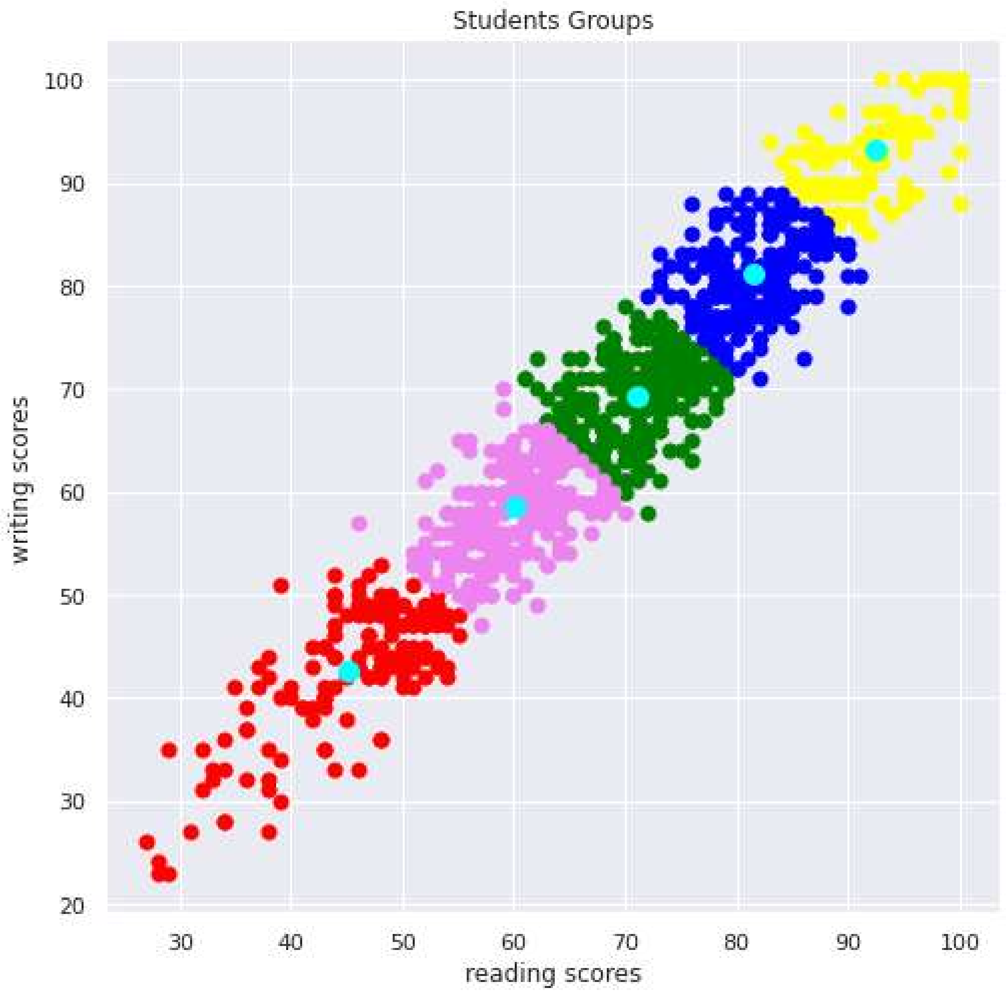


```
#plotting centriods
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='cyan', labe
plt.title('Students Groups')
plt.xlabel('reading score')
plt.ylabel('writing scores')
plt.show()
```



```
#plotting clusters along with centroids
plt.figure(figsize=(8,8))
plt.scatter(X[Y==0,0], X[Y==0,1], s=50, c='green', label='Cluster 1')
plt.scatter(X[Y==1,0], X[Y==1,1], s=50, c='red', label='Cluster 2')
plt.scatter(X[Y==2,0], X[Y==2,1], s=50, c='yellow', label='Cluster 3')
plt.scatter(X[Y==3,0], X[Y==3,1], s=50, c='blue', label='Cluster 4')
plt.scatter(X[Y==4,0], X[Y==4,1], s=50, c='violet', label='Cluster 5')

plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='cyan', labe
plt.title('Students Groups')
plt.xlabel('reading scores')
plt.ylabel('writing scores')
plt.show()
```


Students Groups
writing scores
reading scores