In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [2]:
```
df=pd.read_csv('Customers.csv')
```

In [3]:
```
df.head(15)
```

Out[3]:

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2
4	5	Female	31	38000	40	Entertainment	2	6
5	6	Female	22	58000	76	Artist	0	2
6	7	Female	35	31000	6	Healthcare	1	3
7	8	Female	23	84000	94	Healthcare	1	3
8	9	Male	64	97000	3	Engineer	0	3
9	10	Female	30	98000	72	Artist	1	4
10	11	Male	67	7000	14	Engineer	1	3
11	12	Female	35	93000	99	Healthcare	4	4
12	13	Female	58	80000	15	Executive	0	5
13	14	Female	24	91000	77	Lawyer	1	1
14	15	Male	37	19000	13	Doctor	0	1

In [4]:
```
df.shape
```

Out[4]:
```
(2000, 8)
```

In [5]:
```
df.info
```

Out[5]:
```
<bound method DataFrame.info of       CustomerID  Gender  Age  Annual Income
($)  Spending Score (1-100)  \
0              1    Male   19              15000                     39
1              2    Male   21              35000                     81
2              3  Female   20              86000                      6
3              4  Female   23              59000                     77
4              5  Female   31              38000                     40
...          ...     ...  ...                ...                    ...
1995        1996  Female   71             184387                     40
1996        1997  Female   91              73158                     32
1997        1998    Male   87              90961                     14
1998        1999    Male   77             182109                      4
1999        2000    Male   90             110610                     52

         Profession  Work Experience  Family Size
0        Healthcare                1            4
1          Engineer                3            3
2          Engineer                1            1
3            Lawyer                0            2
4     Entertainment                2            6
...             ...              ...          ...
1995         Artist                8            7
1996         Doctor                7            7
1997     Healthcare                9            2
1998      Executive                7            2
1999  Entertainment                5            2

[2000 rows x 8 columns]>
```

In [6]:
```
df.isnull().sum
```

Out[6]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of       CustomerI
D  Gender    Age  Annual Income ($)  Spending Score (1-100)  \
0          False   False  False              False                    False
1          False   False  False              False                    False
2          False   False  False              False                    False
3          False   False  False              False                    False
4          False   False  False              False                    False
...          ...     ...    ...                ...                      ...
1995       False   False  False              False                    False
1996       False   False  False              False                    False
1997       False   False  False              False                    False
1998       False   False  False              False                    False
1999       False   False  False              False                    False

      Profession  Work Experience  Family Size
0          False            False        False
1          False            False        False
2          False            False        False
3          False            False        False
4          False            False        False
...          ...              ...          ...
1995       False            False        False
1996       False            False        False
1997       False            False        False
1998       False            False        False
1999       False            False        False

[2000 rows x 8 columns]>
```

In [7]:
```
X= df.iloc[:,[3,4]].values
```

In [8]:
```
print(X)
```

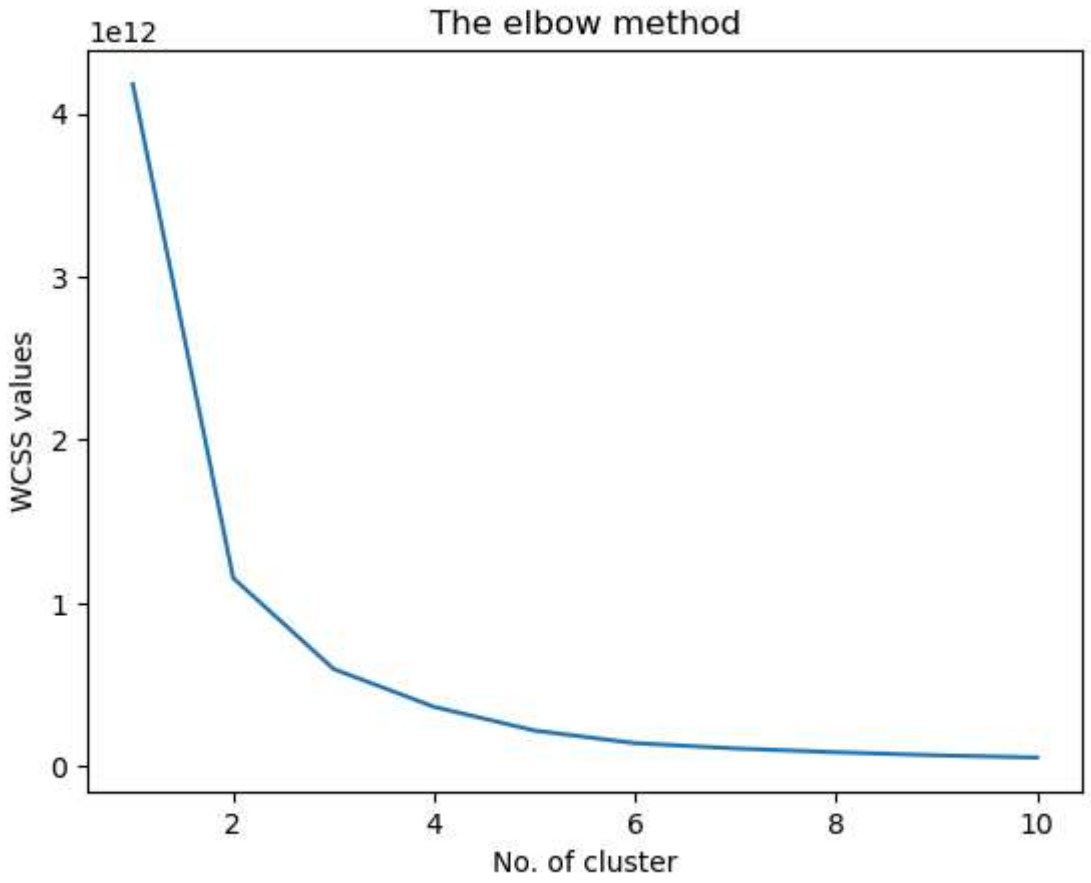
```
[[ 15000     39]
 [ 35000     81]
 [ 86000      6]
 ...
 [ 90961     14]
 [182109      4]
 [110610     52]]
```

In [26]:
```
from sklearn.cluster import KMeans
wcss =[]
```

In [27]:
```
for i in range(1, 11):
  kmeans = skc.KMeans(n_clusters=i, init='k-means++', random_state=0)
  kmeans.fit(X)
  wcss.append(kmeans.inertia_)
```

```
C:\Users\Sridip\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=8.
  warnings.warn(
```

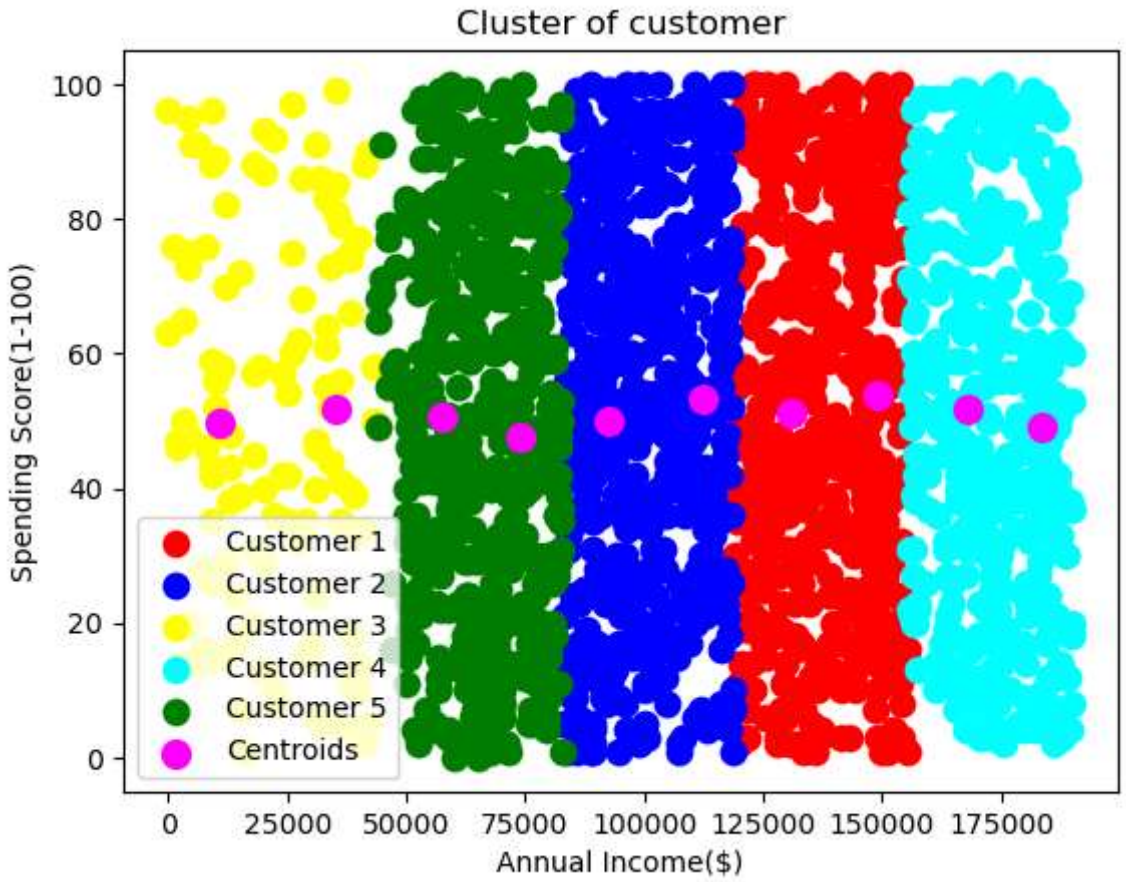
In [28]:
```
plt.plot(range(1,11), wcss)
plt.title('The elbow method')
plt.xlabel('No. of cluster')
plt.ylabel('WCSS values')
plt.show()
```



In [30]:
```
kmeansmodel = KMeans(n_clusters=5, init='k-means++', random_state=0)
```

In [31]:
```
y_kmeans = kmeansmodel.fit_predict(X)
```

In [33]:
```
plt.scatter(X[y_kmeans ==0,0], X[y_kmeans ==0,1], s=80, c="red", label='Custom
plt.scatter(X[y_kmeans ==1,0], X[y_kmeans ==1,1], s=80, c="blue", label='Custo
plt.scatter(X[y_kmeans ==2,0], X[y_kmeans ==2,1], s=80, c="yellow", label='Cus
plt.scatter(X[y_kmeans ==3,0], X[y_kmeans ==3,1], s=80, c="cyan", label='Custo
plt.scatter(X[y_kmeans ==4,0], X[y_kmeans ==4,1], s=80, c="green", label='Cust
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1], s=100,
plt.title('Cluster of customer')
plt.xlabel('Annual Income($)')
plt.ylabel('Spending Score(1-100)')
plt.legend()
plt.show()
```



In [ ]: