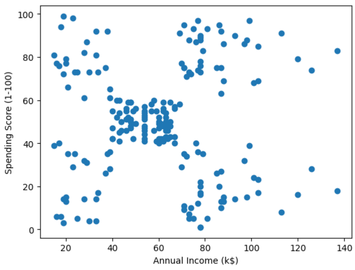
In [1]:
```
import pandas as pd
z=pd.read_csv("C:\\Users\\ravit\\Downloads\\Mall_Customers.csv")
z.drop('CustomerID',axis=1)
z['Gender']=z['Gender'].map({'Male':0,'Female':1})
display(z)
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	0	19	15	39
1	2	0	21	15	81
2	3	1	20	16	6
3	4	1	23	16	77
4	5	1	31	17	40
...	...	...	...	...	...
195	196	1	35	120	79
196	197	1	45	126	28
197	198	0	32	126	74
198	199	0	32	137	18
199	200	0	30	137	83

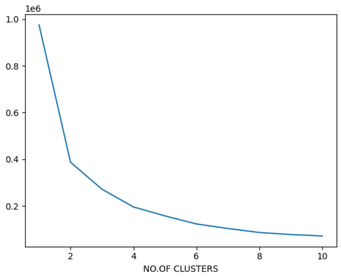
200 rows × 5 columns

In [2]:
```
import matplotlib.pyplot as plt
import pandas as pd
z=pd.read_csv("C:\\Users\\ravit\\Downloads\\Mall_Customers.csv")
z.drop('CustomerID',axis=1)
z['Gender']=z['Gender'].map({'Male':0,'Female':1})
plt.scatter(z['Annual Income (k$)'],z['Spending Score (1-100)'])
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.show()
```



In [3]:
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
    kmeans.fit(z)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.xlabel('NO.OF CLUSTERS')
# we got no.of clusters is 2 as output
```

Out[3]:
```
Text(0.5, 0, 'NO.OF CLUSTERS')
```



In [4]:
```
from sklearn.preprocessing import normalize
import pandas as pd
z=pd.read_csv("C:\\Users\\ravit\\Downloads\\Mall_Customers.csv")
z.drop('CustomerID',axis=1)
z['Gender']=z['Gender'].map({'Male':0,'Female':1})
data_scaled=normalize(z)
data_scaled=pd.DataFrame(data_scaled,columns=z.columns)
display(data_scaled)
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	0.021780	0.000000	0.413826	0.326705	0.849433
1	0.023520	0.000000	0.246956	0.176397	0.952546
2	0.113228	0.037743	0.754851	0.603881	0.226455
3	0.048755	0.012189	0.280342	0.195020	0.938536
4	0.093234	0.018647	0.578052	0.316997	0.745874
...	...	...	...	...	...
195	0.798286	0.004073	0.142551	0.488747	0.321758
196	0.821583	0.004170	0.187671	0.525479	0.116773
197	0.797894	0.000000	0.128953	0.507751	0.298203
198	0.814330	0.000000	0.130948	0.560619	0.073658
199	0.775229	0.000000	0.116284	0.531032	0.321720

200 rows × 5 columns

In [9]:
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import normalize
import matplotlib.pyplot as plt
import pandas as pd
z=pd.read_csv("C:\\Users\\ravit\\Downloads\\Mall_Customers.csv")
z.drop('CustomerID',axis=1)
z['Gender']=z['Gender'].map({'Male':0,'Female':1})
data_scaled=normalize(z)
data_scaled=pd.DataFrame(data_scaled,columns=z.columns)
k_mean=KMeans(n_clusters=2)
k_pre=k_mean.fit_predict(data_scaled)
z['cluster_int']=k_pre
display(z)
```

```
C:\Users\ravit\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

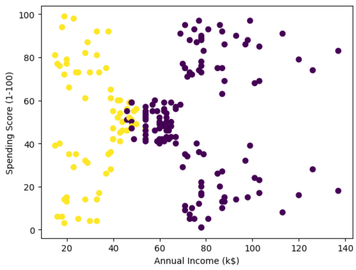
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	cluster_int
0	1	0	19	15	39	0
1	2	0	21	15	81	0
2	3	1	20	16	6	0
3	4	1	23	16	77	0
4	5	1	31	17	40	0
...	...	...	...	...	...	...
195	196	1	35	120	79	1
196	197	1	45	126	28	1
197	198	0	32	126	74	1
198	199	0	32	137	18	1
199	200	0	30	137	83	1

200 rows × 6 columns

In [10]:
```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
z=pd.read_csv("C:\\Users\\ravit\\Downloads\\Mall_Customers.csv")
z.drop('CustomerID',axis=1)
z['Gender']=z['Gender'].map({'Male':0,'Female':1})
k_mean=KMeans(n_clusters=2)
k_pre=k_mean.fit_predict(data_scaled)
z['cluster_int']=k_pre

plt.scatter(z['Annual Income (k$)'],z['Spending Score (1-100)'],c=z['cluster_int'])
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.show()
```

```
C:\Users\ravit\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```


Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
20
40
60
80
100
120
140

In [ ]: