In [42]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import GridSearchCV
```

In [52]:

```
df=pd.read_csv("D:\\python jupyter\data files\customer_segmentation.csv")
```

In [53]:

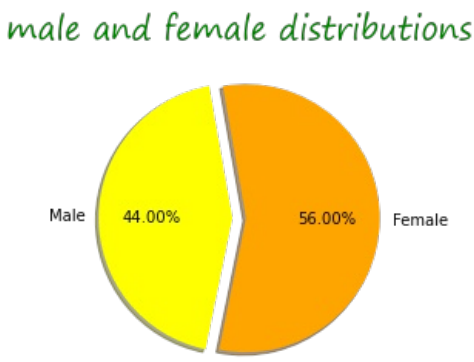
```
df.head()
```

Out[53]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

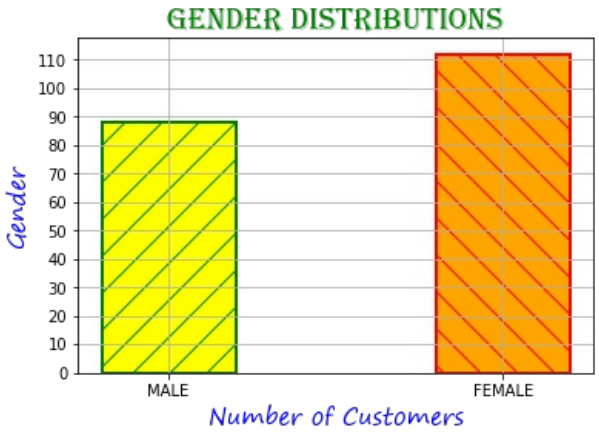
In [54]:

```
male=df.loc[df["Gender"]=="Male"].count()[0]
female=df.loc[df["Gender"]=="Female"].count()[0]
gender=["Male","Female"]
exploding=[0.05,0.05]
plt.pie([male,female],labels=gender,colors=["yellow","orange"],radius=1,startangle=100,autopct="%0.2f%%",explode=
exploding,shadow=True)
plt.title("male and female distributions",fontdict={"fontname":"segoe print","color":"green","fontsize":20})
plt.show()
```



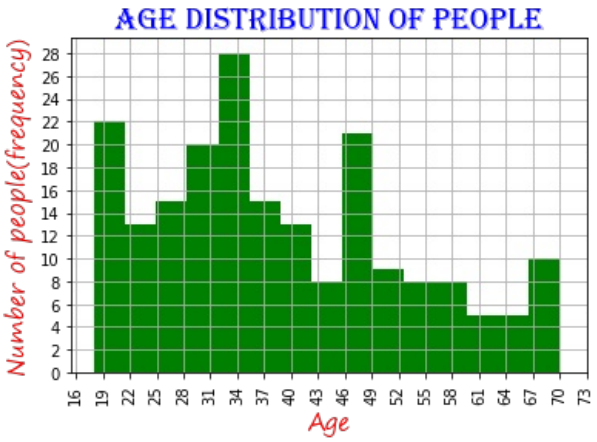
In [55]:

```
labels=["MALE","FEMALE"]
values=[male,female]
designs=["/","\\"]
plt.grid()
plt.yticks(np.arange(0,130,10))
bars=plt.bar(labels,values,color=["yellow","orange"],width=0.4,edgecolor=["green","red"],linewidth=[2,2])
for i in range(len(designs)):
    bars[i].set_hatch(designs[i])
plt.xlabel("Number of Customers",fontname="segoe print",fontsize=15,color="blue")
plt.ylabel("Gender",fontname="segoe print",fontsize=15,color="blue")
plt.title("Gender distributions",fontdict={"fontname":"algerian","color":"green","fontsize":20})
plt.show()
```



In [56]:

```
age=df["Age"]
plt.hist(age,color="green",bins=15)
plt.xticks(np.arange(16,75,3),rotation=90)
plt.yticks(np.arange(0,30,2))
plt.xlabel("Age",fontname="segoe print",fontsize=15,color="red")
plt.ylabel("Number of people(frequency)",fontname="segoe print",fontsize=15,color="red")
plt.title("age distribution of people",fontdict={"fontname":"algerian","color":"blue","fontsize":20})
plt.grid()  #maximum no of people are in the age group of (31-35)
plt.show()
```



In [48]:

```
df.groupby("Gender").mean()
```

Out[48]:

Gender	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Female	97.562500	38.098214	59.250000	51.526786
Male	104.238636	39.806818	62.227273	48.511364

In [57]:

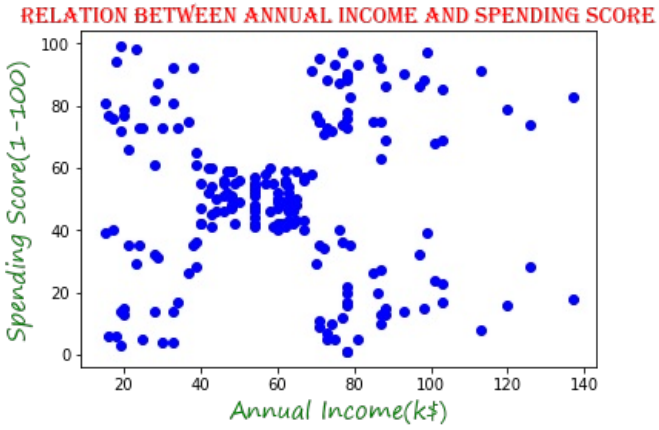
```
#changing male to 0 and female to 1
df["Gender"]=df["Gender"].apply(lambda x: 0 if x=="Male" else 1)
df.head()
```

Out[57]:

	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

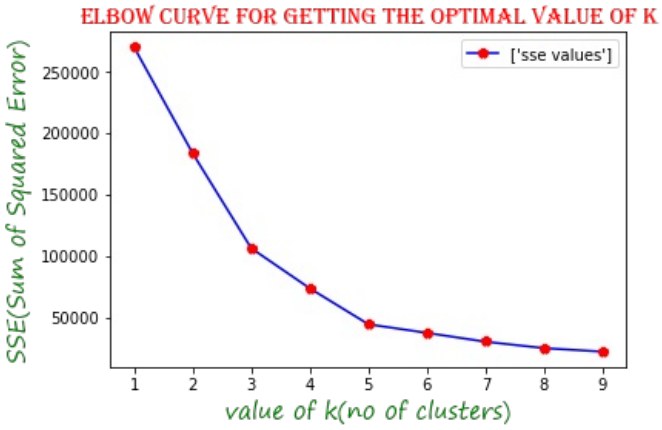
In [78]:

```
#by seeing the scatter plot I am predicting it to be 5 clusters

plt.scatter(df["Annual Income (k$)"],df["Spending Score (1-100)"],color="blue")
plt.xlabel("Annual Income(k$)",fontname="segoe print",fontsize=15,color="green")
plt.ylabel("Spending Score(1-100)",fontname="segoe print",fontsize=15,color="green")
plt.title("Relation between Annual Income and Spending Score",fontdict={"fontname":"algerian","color":"red","font
size":15})
plt.show()
```



In [59]:

```
#checking what show be the optimal K value with the help of elbow_curve
krng=range(1,10)
sse=[]
for k in krng:
    kmc=KMeans(n_clusters=k)
    kmc.fit(df[["Annual Income (k$)","Spending Score (1-100)"]])
    sse.append(kmc.inertia_)
plt.plot(krng,sse,marker="*",color="blue",markeredgecolor="red",markeredgewidth=3,label=["sse values"])

plt.xlabel("value of k(no of clusters)",fontname="segoe print",fontsize=15,color="green")
plt.ylabel("SSE(Sum of Squared Error)",fontname="segoe print",fontsize=15,color="green")
plt.title("Elbow curve for getting the optimal value of k",fontdict={"fontname":"algerian","color":"red","fontsiz
e":15})
plt.legend()# We can see that k=5 is the place where the elbow is folding so we will take that as no.
plt.show()  # of clusters
```



In [60]:

```
#scaling the annual income and spending score column between 0 to 1
"""scalar=MinMaxScaler()
df["Annual Income (k$)"]=scalar.fit_transform(df[["Annual Income (k$)"]])
df["Spending Score (1-100)"]=scalar.fit_transform(df[["Spending Score (1-100)"]])
df.tail()"""
```

Out[60]:

```
'scalar=MinMaxScaler()\ndf["Annual Income (k$)"]=scalar.fit_transform(df[["Annual Income (k$)"]])\nd
f["Spending Score (1-100)"]=scalar.fit_transform(df[["Spending Score (1-100)"]])\ndf.tail()'
```

In [66]:

```
#creating the model and fitting the data into it .
kms=KMeans(n_clusters=5,algorithm="auto")
ypred=kms.fit_predict(df[["Annual Income (k$)","Spending Score (1-100)"]])
df["cluster"]=ypred
```

In [67]:

```
df.head()
```

Out[67]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	cluster
0	1	0	19	15	39	3
1	2	0	21	15	81	4
2	3	1	20	16	6	3
3	4	1	23	16	77	4
4	5	1	31	17	40	3

In [68]:

```
#dividing the clusters into different dataframes
df0=df[df.cluster==0]
df1=df[df.cluster==1]
df2=df[df.cluster==2]
df3=df[df.cluster==3]
df4=df[df.cluster==4]
```

In [69]:

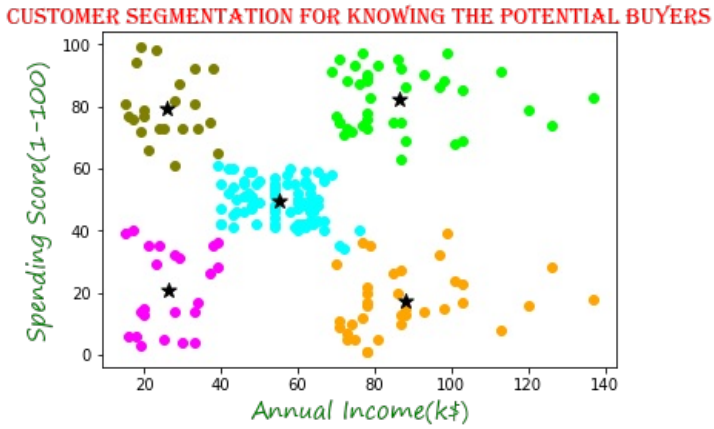
```
# getting the centroids of all the five clusters
centroids=kms.cluster_centers_
centroids
```

Out[69]:

```
array([[86.53846154, 82.12820513],
       [55.2962963 , 49.51851852],
       [88.2       , 17.11428571],
       [26.30434783, 20.91304348],
       [25.72727273, 79.36363636]])
```

In [72]:

```
#plotting all the different clusters with their centroids
plt.scatter(df0["Annual Income (k$)"],df0["Spending Score (1-100)"],color="lime")
plt.scatter(df1["Annual Income (k$)"],df1["Spending Score (1-100)"],color="aqua")
plt.scatter(df2["Annual Income (k$)"],df2["Spending Score (1-100)"],color="orange")
plt.scatter(df3["Annual Income (k$)"],df3["Spending Score (1-100)"],color="fuchsia")
plt.scatter(df4["Annual Income (k$)"],df4["Spending Score (1-100)"],color="olive")
plt.scatter(centroids[:,0],centroids[:,1],color="black",marker="*",s=100)
plt.xlabel("Annual Income(k$)",fontname="segoe print",fontsize=15,color="green")
plt.ylabel("Spending Score(1-100)",fontname="segoe print",fontsize=15,color="green")
plt.title("Customer segmentation for knowing the potential buyers",fontdict={"fontname":"algerian","color":"red",
"fontsize":15})
plt.show()
```



In [ ]: