In [1]:

```
import pandas as pd
```

In [2]:

```
#pip install pandas
```

In [3]:

```
#pip upgrade
```

In [4]:

```
#pip install --upgrade pip
```

In [5]:

```
data = pd.read_csv('E:\mca30\dc.csv')
```

(1) Create dataset

In [6]:

```
data
```

Out[6]:

	sore_throt	fever	swollen_glands	congestion	headache	diagnosis
0	1	1	1	1	1	strep_throat
1	0	0	0	1	1	allergy
2	1	1	0	1	0	cold
3	1	0	1	0	0	strep_throat
4	0	1	0	1	0	cold
5	0	0	0	1	0	allergy
6	0	0	1	0	0	strep_throat
7	1	0	0	1	1	allergy
8	0	1	0	1	1	cold
9	1	1	0	1	1	cold

In [7]:

```
#pip install sklearn
```

In [8]:

```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (1.
2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (f
rom scikit-learn) (1.24.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (fr
om scikit-learn) (1.10.0)
Requirement already satisfied: joblib>=1.1.1 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (f
rom scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\appdata\local\programs\python\python311\lib\site-pack
ages (from scikit-learn) (3.1.0)
Note: you may need to restart the kernel to use updated packages.
```

(2) Implement DecisionTree

In [9]:

```
import sklearn
```

In [10]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [11]:

```
x = data.drop(columns=('diagnosis'))
```

In [12]:

```
x
```

Out[12]:

	sore_throt	fever	swollen_glands	congestion	headache
0	1	1	1	1	1
1	0	0	0	1	1
2	1	1	0	1	0
3	1	0	1	0	0
4	0	1	0	1	0
5	0	0	0	1	0
6	0	0	1	0	0
7	1	0	0	1	1
8	0	1	0	1	1
9	1	1	0	1	1

In [13]:

```
y = data['diagnosis']
```

In [14]:

```
y
```

Out[14]:

```
0    strep_throat
1         allergy
2            cold
3    strep_throat
4            cold
5         allergy
6    strep_throat
7         allergy
8            cold
9            cold
Name: diagnosis, dtype: object
```

In [15]:

```
model = DecisionTreeClassifier()
```

In [16]:

```
model.fit(x,y)
```

Out[16]:

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

(3) Print DecisionTree

In [ ]:

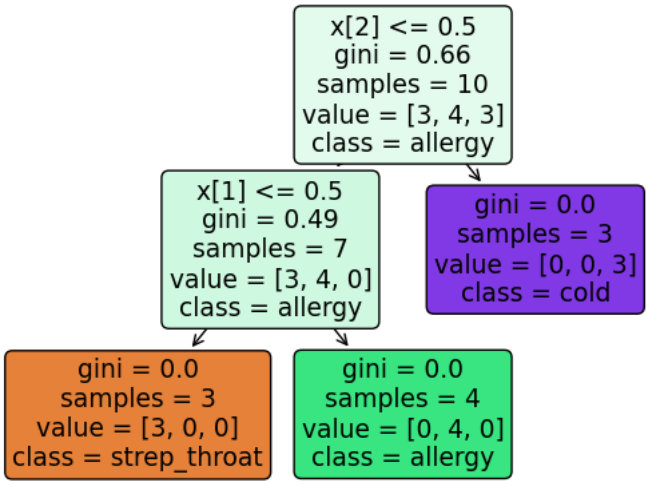
```
from sklearn import tree
```

In [93]:

```
tree.plot_tree(model,filled=True,class_names=y,rounded=True)
```

Out[93]:

```
[Text(0.6, 0.8333333333333334, 'x[2] <= 0.5\ngini = 0.66\nsamples = 10\nvalue = [3, 4, 3]\nclass = allergy'),
 Text(0.4, 0.5, 'x[1] <= 0.5\ngini = 0.49\nsamples = 7\nvalue = [3, 4, 0]\nclass = allergy'),
 Text(0.2, 0.16666666666666666, 'gini = 0.0\nsamples = 3\nvalue = [3, 0, 0]\nclass = strep_throat'),
 Text(0.6, 0.16666666666666666, 'gini = 0.0\nsamples = 4\nvalue = [0, 4, 0]\nclass = allergy'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = cold')]
```



(4) Classifcation of data (0,0,1,0,0)

In [17]:

```
pred = model.predict([[0,0,1,0,0]])
```

```
C:\Users\Admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:409: UserWarning: X does not have
valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

In [18]:

```
print(pred)
```

```
['strep_throat']
```

In [19]:

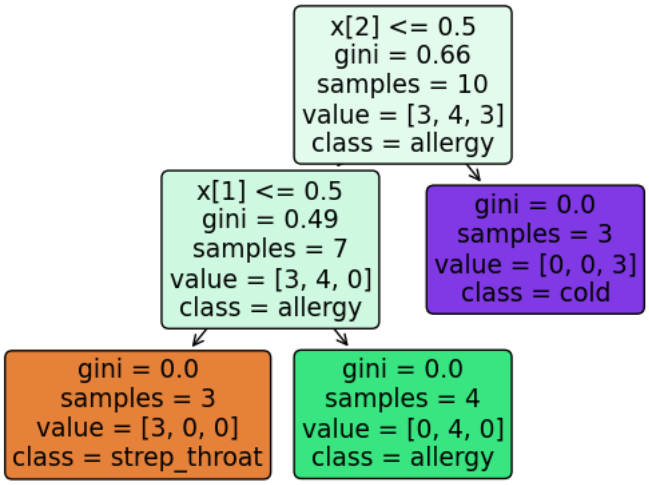
In [20]:

In [92]:

```
```

Out[92]:

```
[Text(0.6, 0.8333333333333334, 'x[2] <= 0.5\ngini = 0.66\nsamples = 10\nvalue = [3, 4, 3]\nclass = allergy'),
 Text(0.4, 0.5, 'x[1] <= 0.5\ngini = 0.49\nsamples = 7\nvalue = [3, 4, 0]\nclass = allergy'),
 Text(0.2, 0.16666666666666666, 'gini = 0.0\nsamples = 3\nvalue = [3, 0, 0]\nclass = strep_throat'),
 Text(0.6, 0.16666666666666666, 'gini = 0.0\nsamples = 4\nvalue = [0, 4, 0]\nclass = allergy'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = cold')]
```



In [ ]:

```
```

B) Bar Plot

In [43]:

```
men = [22,30,35,35,26]
women= [25,32,30,35,29]
```

In [70]:

```
interval = [10,20,30,40,50]
interval1 = [12,22,32,42,52]
```

In [50]:

```
import matplotlib.pyplot as plt
```

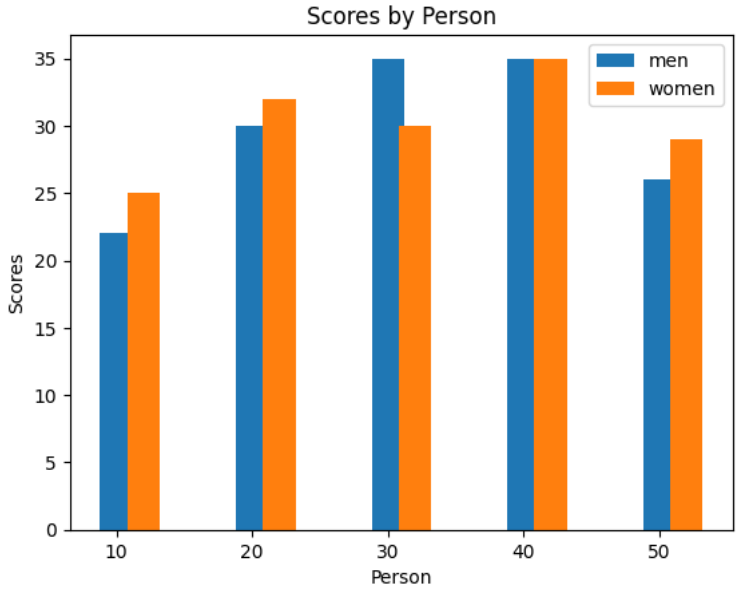
In [76]:

```
plt.bar(interval,men,width=2.4,label='men')
plt.bar(interval1,women,width=2.4,label='women')

plt.legend()
plt.xlabel('Person')
plt.ylabel('Scores')
plt.title('Scores by Person')
```

Out[76]:

```
Text(0.5, 1.0, 'Scores by Person')
```



B) (ii) Pie Chart

In [79]:

```
data2 = pd.read_csv("E:\mca30\medal.csv")

data2
```

Out[79]:

	country	gold_medal
0	United States	46
1	Great Britain	27
2	China	26
3	Russia	19
4	Germany	17

In [82]:

```
country = data2['country']
country
```

Out[82]:

```
0    United States
1    Great Britain
2            China
3           Russia
4          Germany
Name: country, dtype: object
```

In [84]:

```
gold_medal =data2['gold_medal']
gold_medal
```

Out[84]:

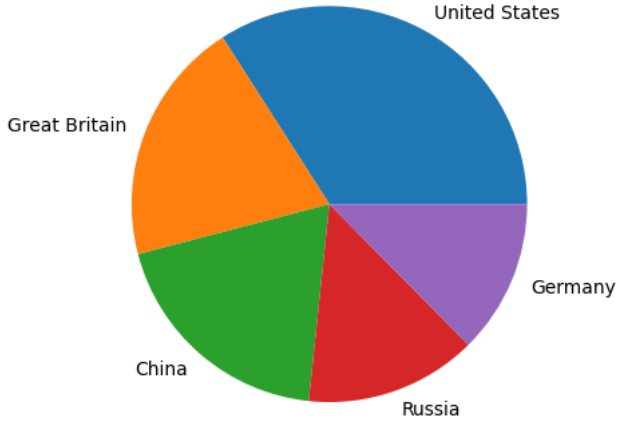
```
0    46
1    27
2    26
3    19
4    17
Name: gold_medal, dtype: int64
```

In [87]:

```
plt.pie(gold_medal,labels=country)
```

Out[87]:

```
([<matplotlib.patches.Wedge at 0x22c6f20f9d0>,
  <matplotlib.patches.Wedge at 0x22c6f2ae650>,
  <matplotlib.patches.Wedge at 0x22c6f2afd10>,
  <matplotlib.patches.Wedge at 0x22c6f3dfd90>,
  <matplotlib.patches.Wedge at 0x22c6f1607d0>],
 [Text(0.5276844259140197, 0.9651679370181085, 'United States'),
  Text(-1.0246277154553132, 0.40017251869765635, 'Great Britain'),
  Text(-0.7168229424450668, -0.8343649496379847, 'China'),
  Text(0.3641697549943344, -1.037969358674603, 'Russia'),
  Text(1.015038685557071, -0.4239062005002681, 'Germany')])
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: