In [1]:
```
# import the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# load the dataset
df=pd.read_csv('D:/Downloads/UM/Heart Disease data.csv')
df
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

In [3]:
```
# check the brief info
df.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [4]:
```
# check the buttom 5 record
df.tail()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

In [5]:
```
# shape of the dataset
df.shape
```

Out[5]:
```
(1025, 14)
```

In [6]:
```
# check the basic info of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Attribute Information:

- age
- sex
- chest pain type (4 values)
- resting blood pressure
- serum cholestoral in mg/dl
- fasting blood sugar > 120 mg/dl
- resting electrocardiographic results (values 0,1,2)
- maximum heart rate achieved
- exercise induced angina
- oldpeak = ST depression induced by exercise relative to rest
- the slope of the peak exercise ST segment
- number of major vessels (0-3) colored by flourosopy
- thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

In [7]:
```
# check the sex types
df['sex'].value_counts()
```

Out[7]:
```
1    713
0    312
Name: sex, dtype: int64
```

In [8]:
```
# check the cp types and their value counts
df['cp'].value_counts()
```

Out[8]:
```
0    497
2    284
1    167
3     77
Name: cp, dtype: int64
```

In [9]:
```
# check the fasting blood sugar types and value counts
df['fbs'].value_counts()
```

Out[9]:
```
0    872
1    153
Name: fbs, dtype: int64
```

In [10]:
```
# check the resting electrocardiographic results value counts
df['restecg'].value_counts()
```

Out[10]:
```
1    513
0    497
2     15
Name: restecg, dtype: int64
```

In [11]:
```
#
df['exang'].value_counts()
```

Out[11]:
```
0    680
1    345
Name: exang, dtype: int64
```

In [12]:
```
df['slope'].value_counts()
```

Out[12]:
```
1    482
2    469
0     74
Name: slope, dtype: int64
```

In [13]:
```
df['ca'].value_counts()
```

Out[13]:
```
0    578
1    226
2    134
3     69
4     18
Name: ca, dtype: int64
```

In [14]:
```
df['thal'].value_counts()
```

Out[14]:
```
2    544
3    410
1     64
0      7
Name: thal, dtype: int64
```

In [15]:
```
df['target'].value_counts()
```

Out[15]:
```
1    526
0    499
Name: target, dtype: int64
```

In [16]:
```
# check how many features are not null in our dataset
df.notnull().sum()
```

Out[16]:
```
age         1025
sex         1025
cp          1025
trestbps    1025
chol        1025
fbs         1025
restecg     1025
thalach     1025
exang       1025
oldpeak     1025
slope       1025
ca          1025
thal        1025
target      1025
dtype: int64
```

In [17]:
```
# check is there any null values in our dataset
df.isnull().sum()
```

Out[17]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

There is no null values

In [18]:
```
# check the duplicated record
df.duplicated().sum()
```

Out[18]:
```
723
```

There is a 723 duplicated records so simply remove the duplicates

In [19]:
```
# there is no use of duplicated values so simply drop the duplicated values
df.drop_duplicates(inplace=True,ignore_index=True)
```

In [20]:
```
# after removing the duplicated records check the dataset
df
```

Out[20]:

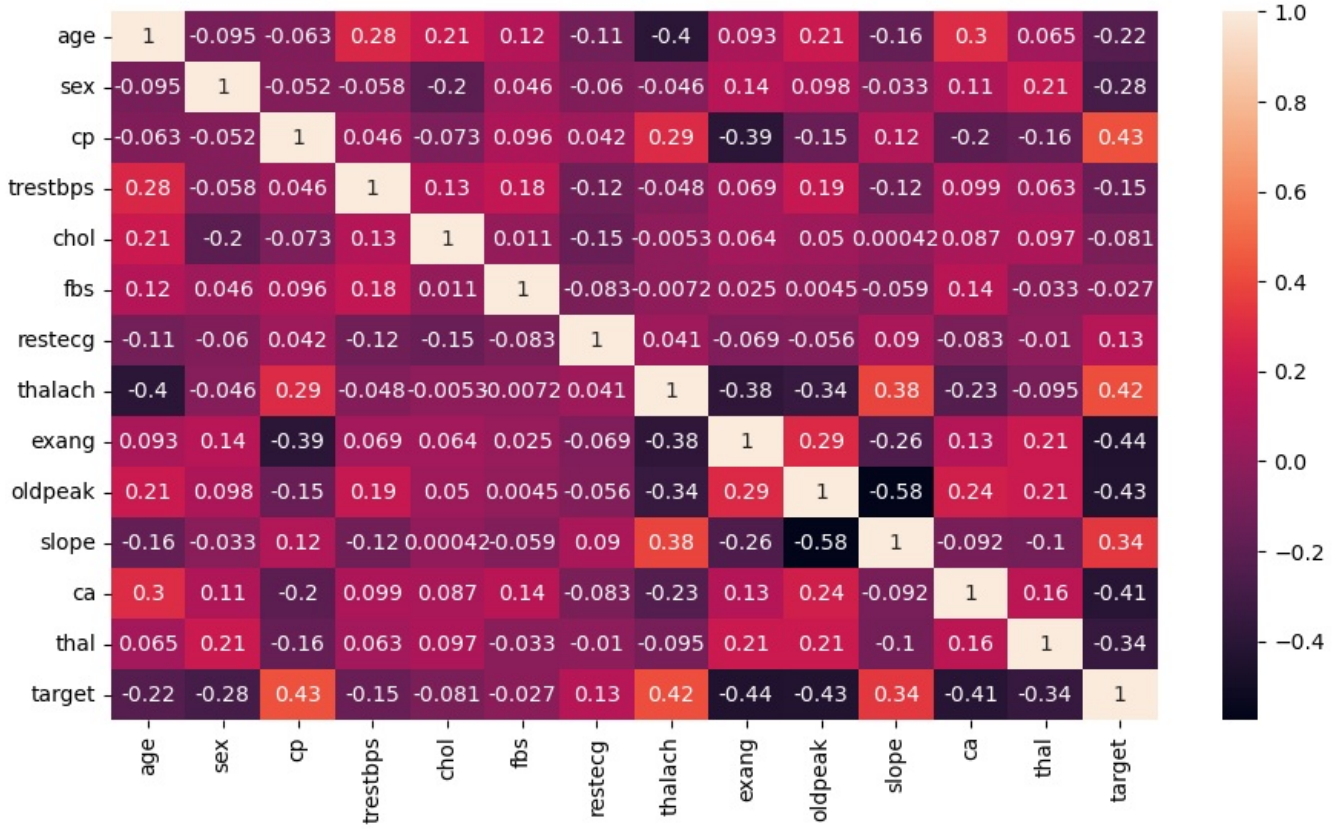
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
297	68	0	2	120	211	0	0	115	0	1.5	1	0	2	1
298	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
299	52	1	0	128	255	0	1	161	1	0.0	2	1	3	0
300	59	1	3	160	273	0	0	125	0	0.0	2	0	2	0
301	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

302 rows × 14 columns

In [21]:
```
# shape of the dataset after removing the duplicated records
df.shape
```

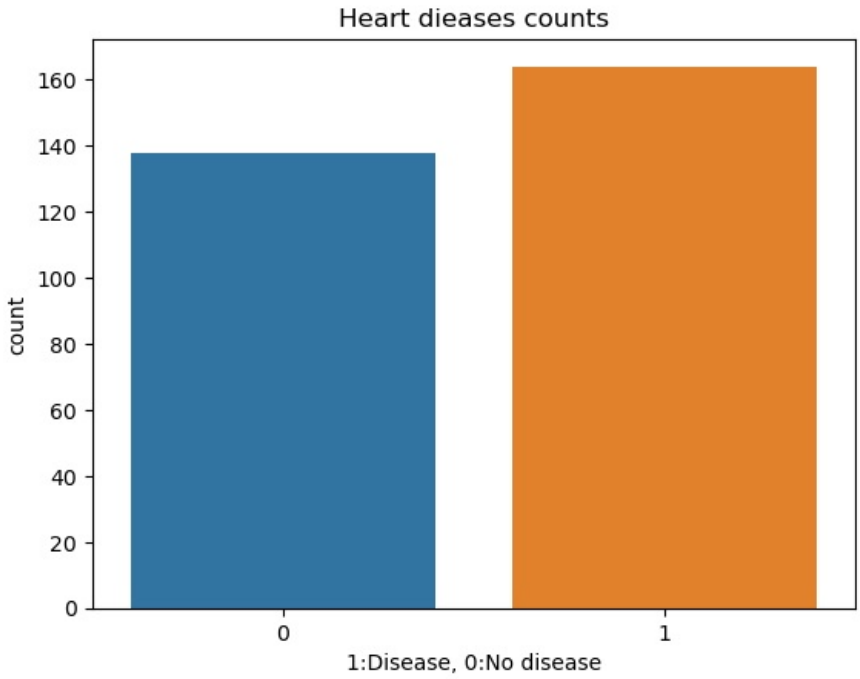
Out[21]:
```
(302, 14)
```

In [22]:
```
# correlation between the features
plt.figure(figsize=(11,6))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



In [23]:
```
# sns.pairplot(df,hue='target')
# plt.show()
```

In [24]:
```
sns.countplot(x=df['target'])
plt.title('Heart dieases counts')
plt.xlabel('1:Disease, 0:No disease')
plt.show()
```



In [26]:
```
# after cleaning the dataset converted into new dataset
df.to_csv('Heart_Disease_Data.csv',index=False)
```