In [70]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [71]:
```
## Loading the data from csv file too a panda Dataframe
titanic_data = pd.read_csv(r"C:\Users\Kamal Kant\OneDrive\Documents\Titanic-Dat
```

In [72]:
```
titanic_data.head()
```

Out[72]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	N

In [73]:
```
# checking the number of columns and column in data frame
titanic_data.shape
```

Out[73]:
```
(891, 12)
```

In [74]:

```
# getting some information about the data
titanic_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [75]:

```
# check the number of missing values in each column
titanic_data.isnull().sum()
```

Out[75]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [76]:

```
# drop the Cabin column from dataframe
titanic_data = titanic_data.drop(columns='Cabin', axis=1)
```

In [77]:

```
titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

In [78]:

```
# finding the mod value of embarked column
print(titanic_data['Embarked'].mode())
```

```
0    S
Name: Embarked, dtype: object
```

In [79]:

```
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=Tru
```

In [80]:

```
# check the number of missing values in each column
titanic_data.isnull().sum()
```

Out[80]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [81]:

```
titanic_data.describe()
```

Out[81]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [82]:

```
titanic_data['Survived'].value_counts()
```

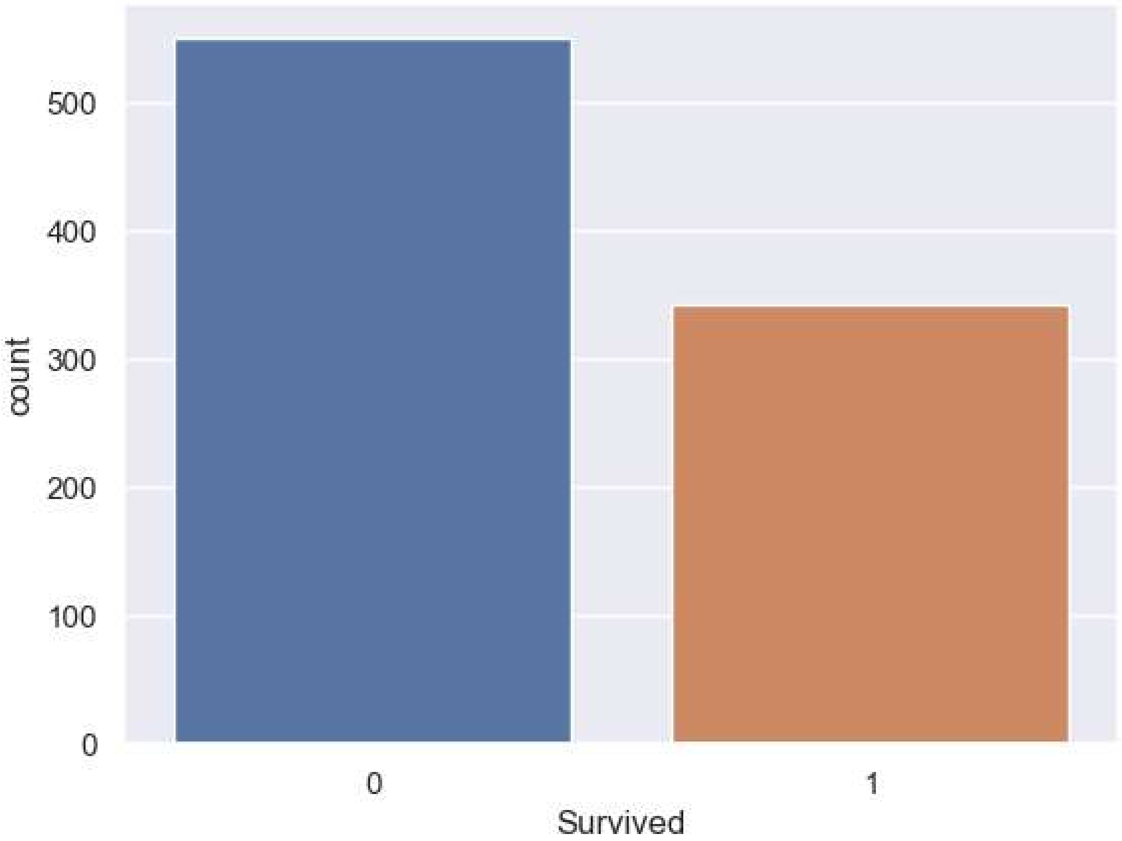
Out[82]:

```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [83]:

```
sns.set()
```

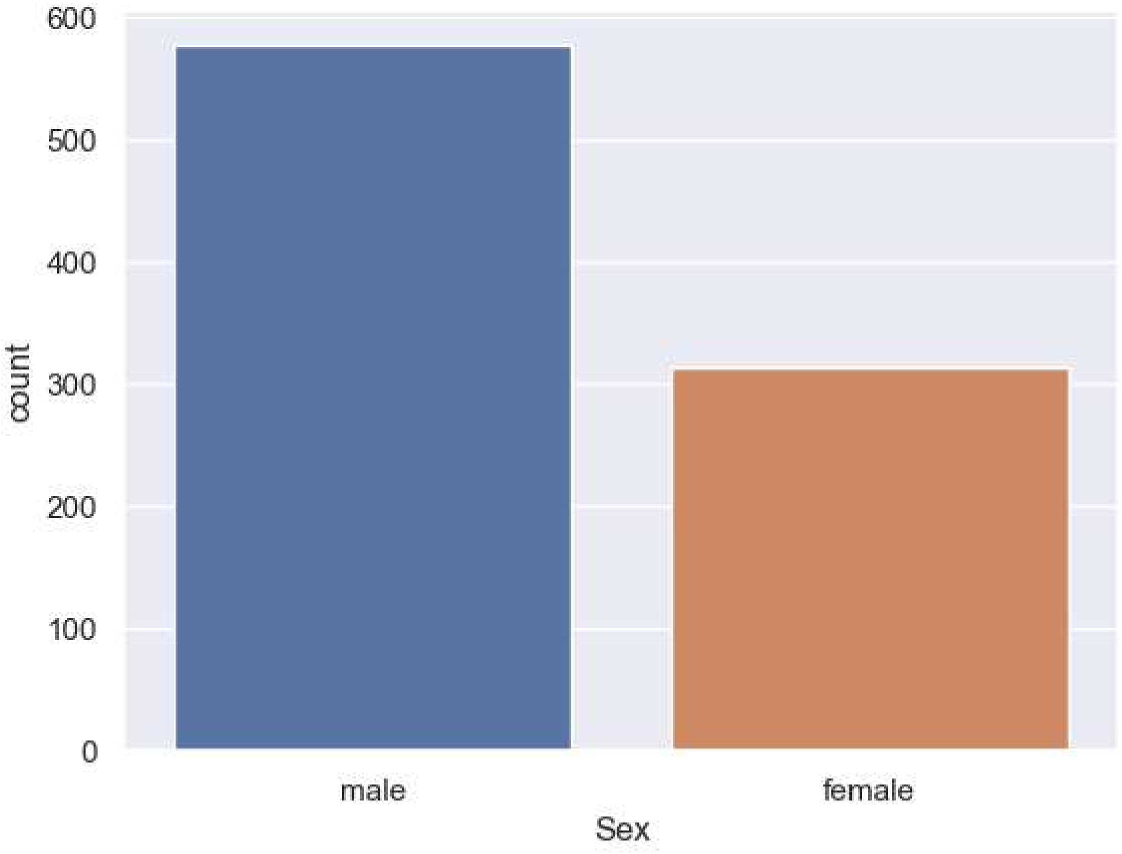
In [84]:
```
sns.countplot(x='Survived',data=titanic_data)
plt.show()
```



In [85]:
```
titanic_data['Sex'].value_counts()
```

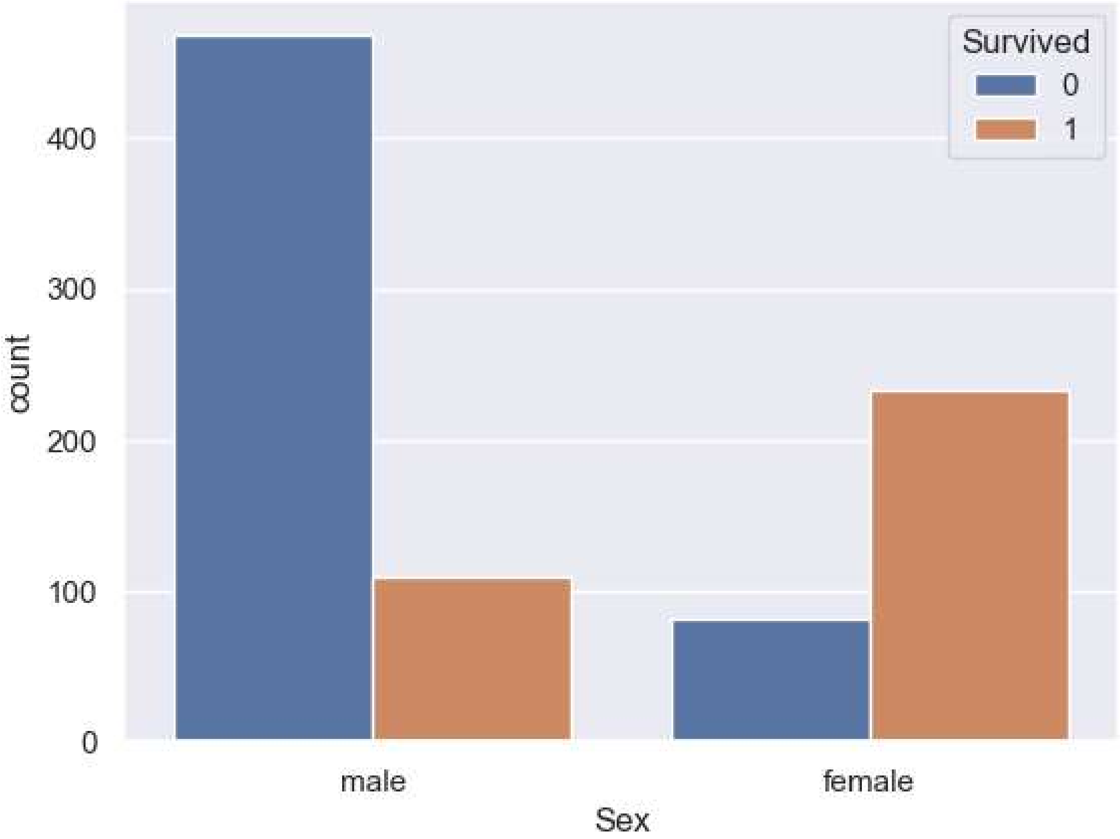
Out[85]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

In [86]:
```
sns.countplot(x='Sex',data=titanic_data)
plt.show()
```



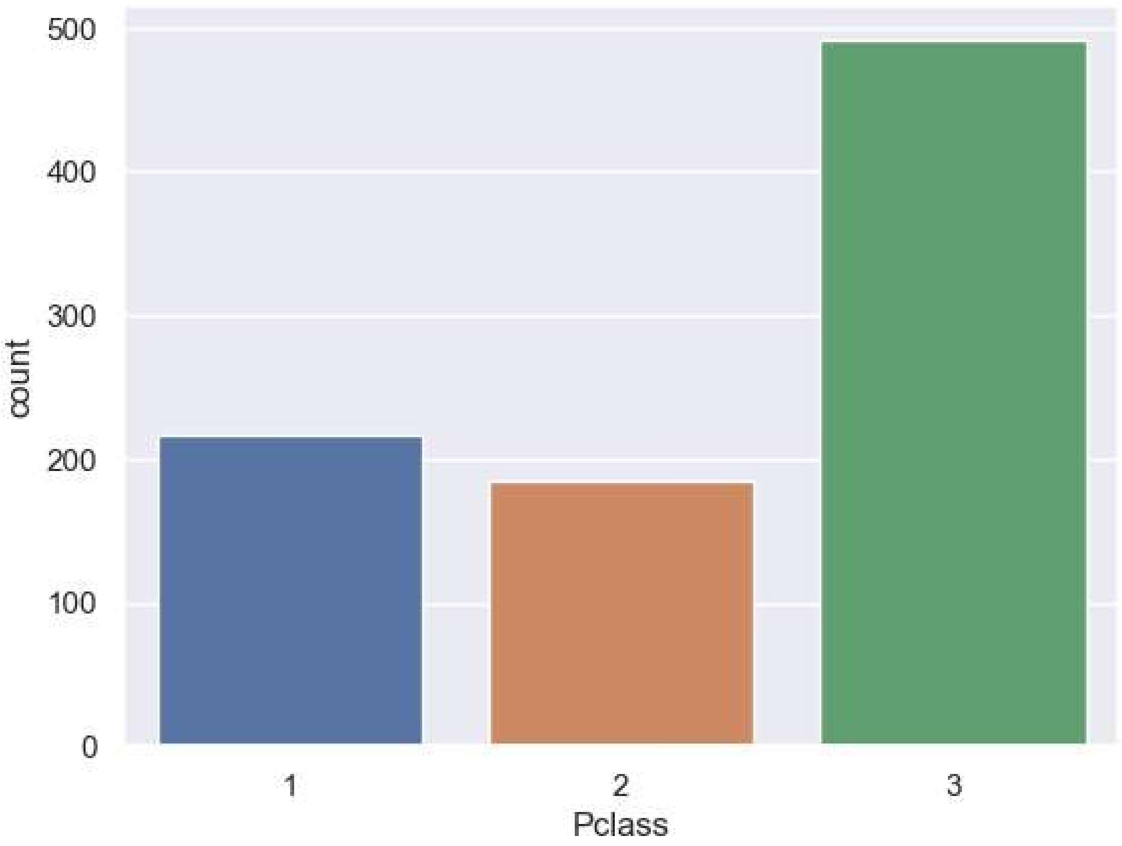
```
In [87]: sns.countplot(x='Sex', hue='Survived', data=titanic_data)
```

```
Out[87]: <Axes: xlabel='Sex', ylabel='count'>
```



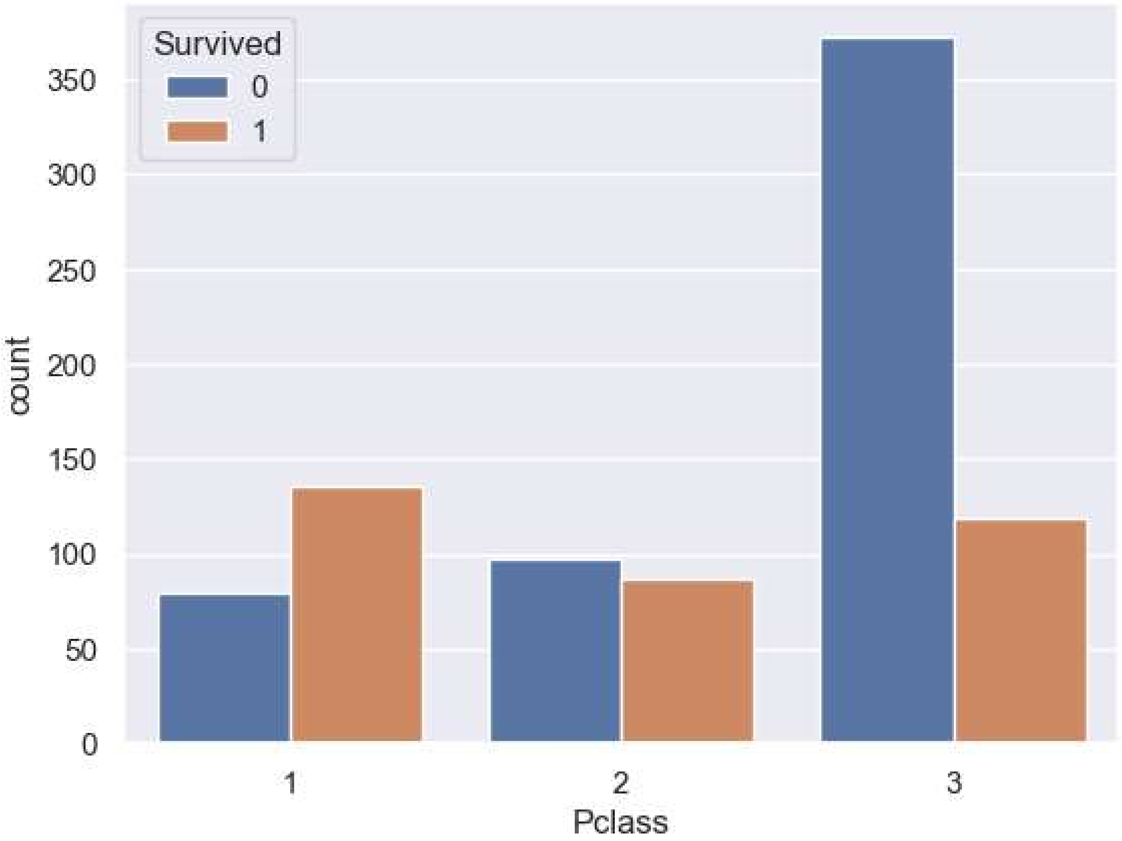
In [88]:
```
sns.countplot(x='Pclass',data=titanic_data)
```

Out[88]:
```
<Axes: xlabel='Pclass', ylabel='count'>
```



In [89]:
```
sns.countplot(x='Pclass', hue='Survived', data=titanic_data)
```

Out[89]:
```
<Axes: xlabel='Pclass', ylabel='count'>
```



In [90]:
```
titanic_data['Sex'].value_counts()
```

Out[90]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

In [91]:
```
titanic_data['Embarked'].value_counts()
```

Out[91]:
```
Embarked
S    646
C    168
Q     77
Name: count, dtype: int64
```

In [92]:
```
titanic_data.replace({'Sex':{'male':0, 'female':1}, 'Embarked':{'S':0,'C':1,'Q'
```

In [93]:
```
titanic_data.head()
```

Out[93]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emba
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	

In [94]:
```
#Seprating the features and target
x = titanic_data.drop(columns = ['PassengerId', 'Name', 'Ticket', 'Survived'],a
y = titanic_data['Survived']
```

In [95]:
```
print(x)
```

```
     Pclass  Sex        Age  SibSp  Parch     Fare  Embarked
0         3    0  22.000000      1      0   7.2500         0
1         1    1  38.000000      1      0  71.2833         1
2         3    1  26.000000      0      0   7.9250         0
3         1    1  35.000000      1      0  53.1000         0
4         3    0  35.000000      0      0   8.0500         0
..      ...  ...        ...    ...    ...      ...       ...
886       2    0  27.000000      0      0  13.0000         0
887       1    1  19.000000      0      0  30.0000         0
888       3    1  29.699118      1      2  23.4500         0
889       1    0  26.000000      0      0  30.0000         1
890       3    0  32.000000      0      0   7.7500         2

[891 rows x 7 columns]
```

In [96]:
```
print(y)
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

In [97]:
```
# Spliting the data into traning data and test data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random
```

In [98]:
```
print(x.shape)
print(x_train.shape)
print(x_test.shape)
```

```
(891, 7)
(712, 7)
(179, 7)
```

In [99]:
```
# Traning model
# Logistic Regression
```

In [100]:
```
model = LogisticRegression(max_iter=1000)
model.fit(x_train, y_train)
```

Out[100]:
```
LogisticRegression(max_iter=1000)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [101]:
```
#Accuracy of traning data
x_train_prediction = model.predict(x_train)
```

```
print(x_train_prediction)
```

In [102]:
```
traning_data_accuracy = accuracy_score(y_train, x_train_prediction)
print('Accuracy score of Traning data :',traning_data_accuracy )
```

```
Accuracy score of Traning data : 0.8089887640449438
```

In [103]:
```
# Accuracy on test data
x_test_prediction = model.predict(x_test)
```

In [104]:
```
print(x_test_prediction)
```

```
[0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0
 1 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 0 0
 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0 0
 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0]
```

In [ ]: