```
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
```

```
titanic_data = pd.read_csv("/content/train.csv")
```

```
titanic_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0

```
titanic_data.shape
```

```
(891, 12)
```

```
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
```

```
titanic_data.isnull().sum()
```

```
                 0
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
```

Double-click (or enter) to edit

```
titanic_data = titanic_data.drop(columns='Cabin', axis=1)
```

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

```
titanic_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 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  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

```
titanic_data.isnull().sum()
```

```
               0
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       2

dtype: int64
```

```
print (titanic_data['Embarked'].mode())
```

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

```
print(titanic_data['Embarked'].mode()[0])
```

```
S
```

```
Start coding or generate with AI.
```

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

```
titanic_data.isnull().sum()
```

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

dtype: int64
```

```
titanic_data.describe()
```

```
       PassengerId    Survived      Pclass         Age       SibSp       Parch         Fa
count   891.000000  891.000000  891.000000  891.000000  891.000000  891.000000  891.000
 mean   446.000000    0.383838    2.308642   29.699118    0.523008    0.381594   32.204
  std   257.353842    0.486592    0.836071   13.002015    1.102743    0.806057   49.693
  min     1.000000    0.000000    1.000000    0.420000    0.000000    0.000000    0.000
  25%   223.500000    0.000000    2.000000   22.000000    0.000000    0.000000    7.910
  50%   446.000000    0.000000    3.000000   29.699118    0.000000    0.000000   14.454
  75%   668.500000    1.000000    3.000000   35.000000    1.000000    0.000000   31.000
  max   891.000000    1.000000    3.000000   80.000000    8.000000    6.000000  512.329
```

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

```
          count
Survived
       0    549
       1    342

dtype: int64
```

```
sns.set()
```

```
sns.countplot(titanic_data['Survived'])
```

```
<Axes: ylabel='count'>
```

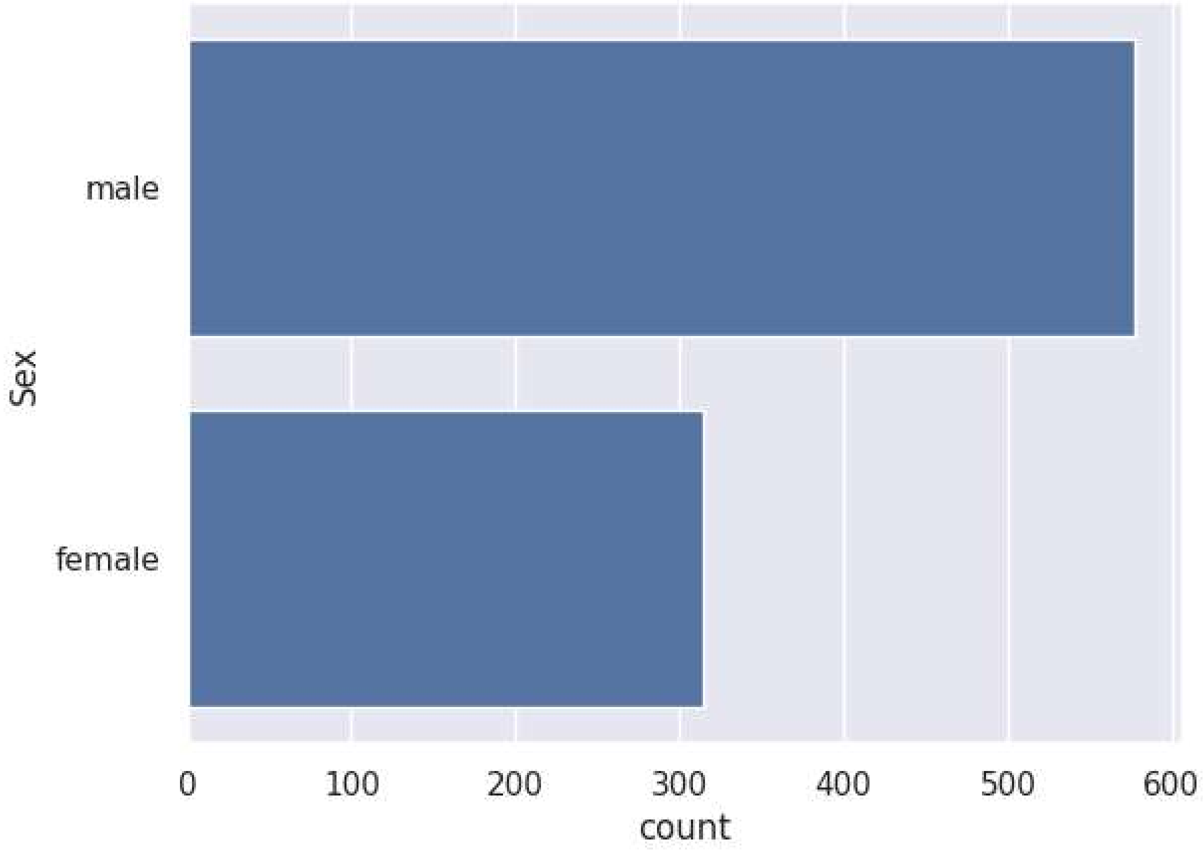
```
titanic_data['Sex'].value_counts()
```

```
         count
Sex
male       577
female     314

dtype: int64
```

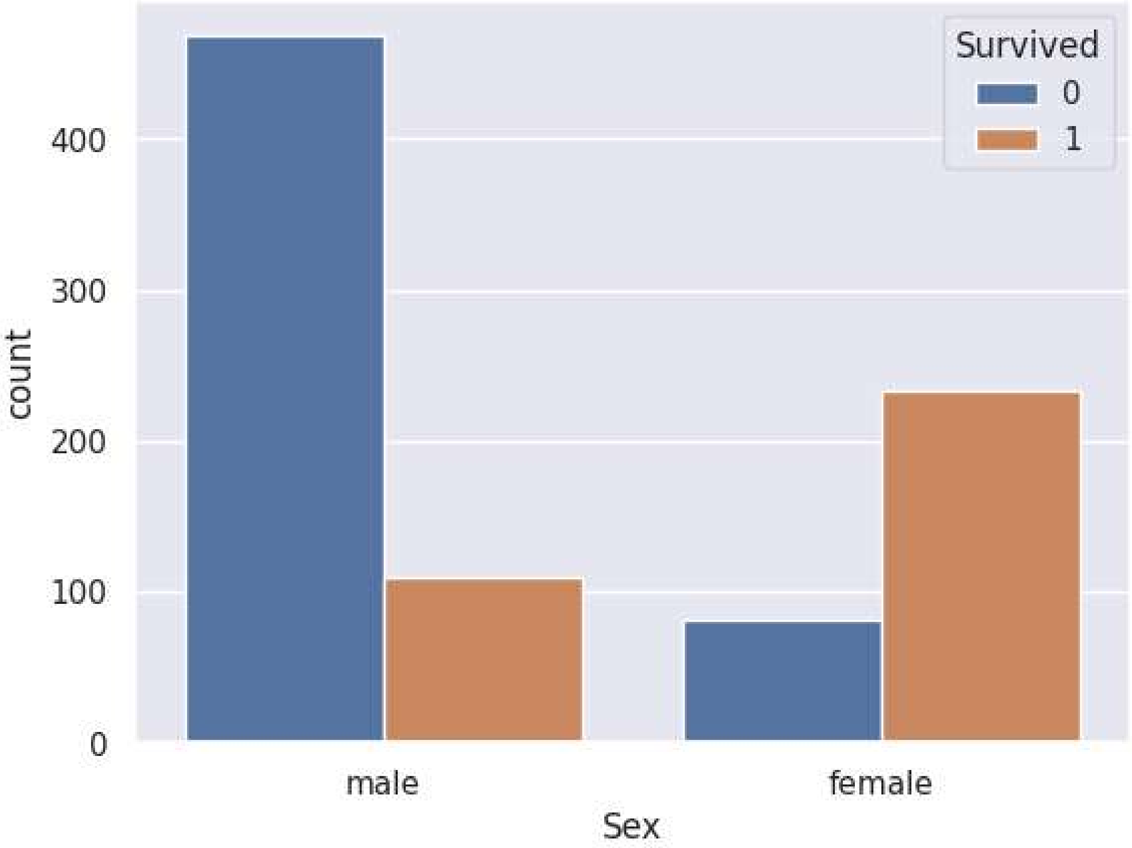
```
sns.countplot(titanic_data['Sex'])
```

```
<Axes: xlabel='count', ylabel='Sex'>
```



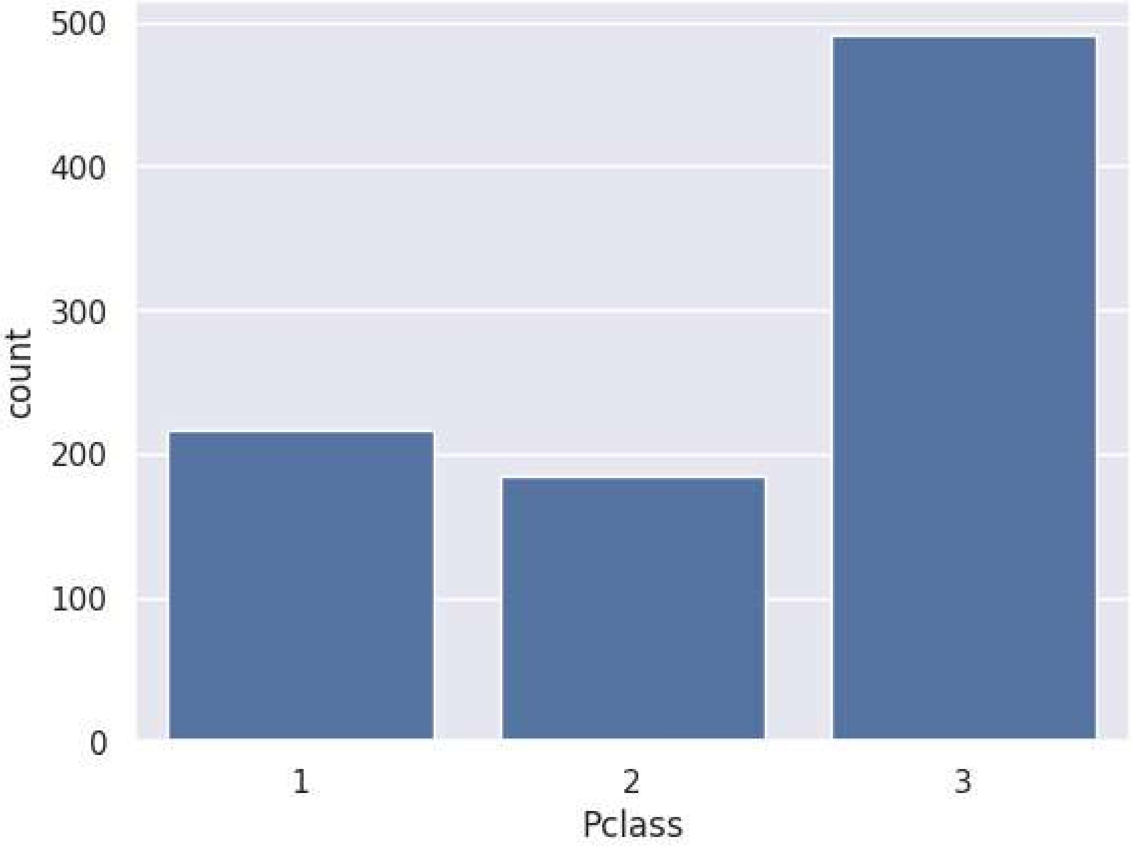
```
sns.countplot(x='Sex', hue='Survived', data=titanic_data)
```

```
<Axes: xlabel='Sex', ylabel='count'>
```



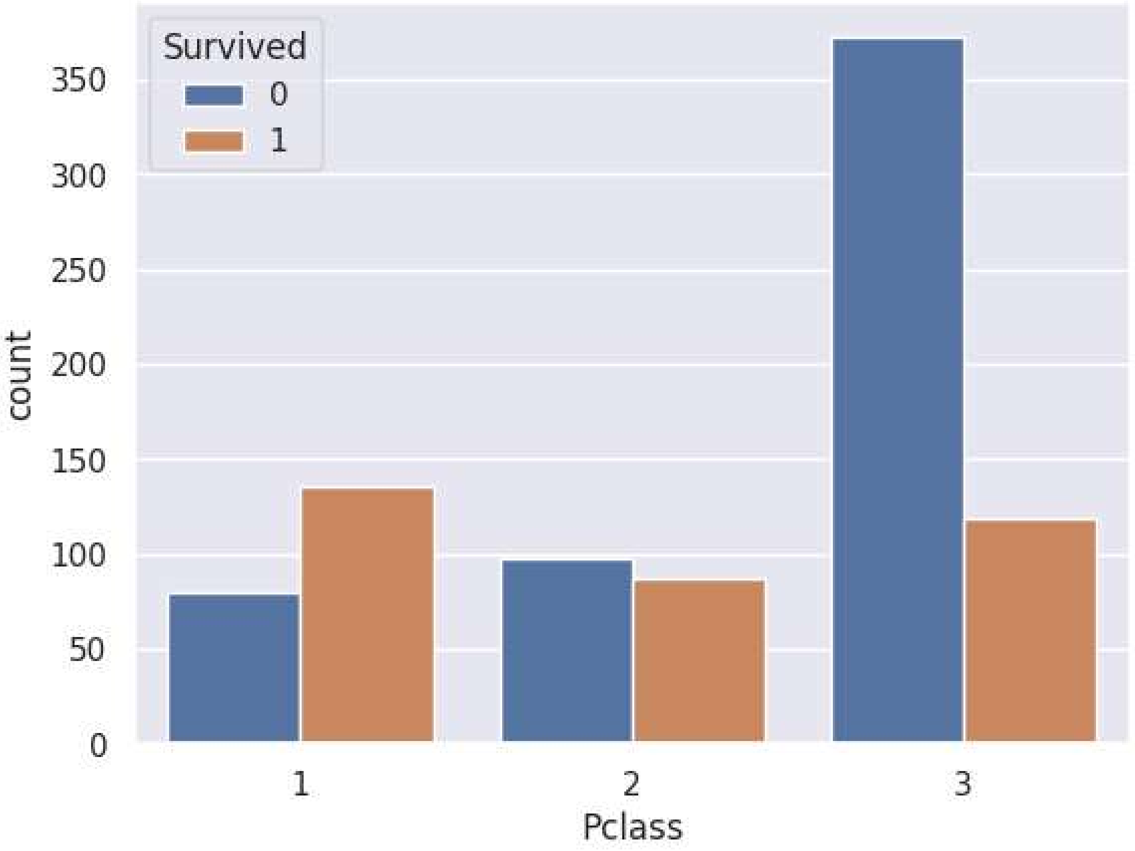
```
sns.countplot(x='Pclass', data=titanic_data)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```



```
sns.countplot(x='Pclass', hue='Survived', data=titanic_data)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```



Encode categorical columms/data

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

```
          count
   Sex
  male      577
 female     314

dtype: int64
```

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

```
            count
Embarked
    S         646
    C         168
    Q          77

dtype: int64
```

```
titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}}, inpla
```

```
titanic_data.replace()
```

```
<ipython-input-53-431163847962>:1: FutureWarning: DataFrame.replace without 'value' a
  titanic_data.replace()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	0	22.000000	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.000000	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	1	26.000000	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.000000	1	0	113803
4	5	0	3	Allen, Mr. William Henry	0	35.000000	0	0	373450
...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	0	27.000000	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	1	19.000000	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	1	29.699118	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	0	26.000000	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	0	32.000000	0	0	370376

891 rows × 11 columns

```
x = titanic_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)
y = titanic_data['Survived']
```

```
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]
```

```
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
```

split datainto test data and train data

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=2)
```

```
print(x.shape, x_train.shape, x_test.shape)
```

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

logistical regerssion and model traning

```
model = LogisticRegression()
```

```
#use the train data
model.fit(x_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
▾ LogisticRegression
LogisticRegression()
```

```
x_train_prediction = model.predict(x_train)
```

```
print(x_train_prediction)
```

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