EDS MINI PROJECT

TITLE OF PROJECT : IPL 2022

GROUP MEMBERS:-

1) 516- Sai Suresh Gulve
2) 505- Shantanu Bhalerao
3) 521- Harshad Jadhav

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('IPL_Matches_2022.csv')
```

```
total_matches = df.shape[0]
print("Total matches played:", total_matches)
```

```
Total matches played: 74
```

```
teams = df["Team1"].unique()
print("Teams participated:", teams)
```

```
Teams participated: ['Rajasthan Royals' 'Royal Challengers Bangalore'
'Sunrisers Hyderabad'
 'Delhi Capitals' 'Chennai Super Kings' 'Gujarat Titans'
 'Lucknow Super Giants' 'Kolkata Knight Riders' 'Punjab Kings'
 'Mumbai Indians']
```

```
most_wins = df["WinningTeam"].value_counts().idxmax()
print("Team with the most wins:", most_wins)
```

```
Team with the most wins: Gujarat Titans
```

```
team_name = "Royal Challengers Bangalore"
team_wins = df[df["WinningTeam"] == team_name].shape[0]
print(team_name, "won", team_wins, "matches.")
```

```
Royal Challengers Bangalore won 9 matches.
```

```
winning_team = df["WinningTeam"].value_counts().idxmax()
print("Winner of IPL 2022:", winning_team)
```

```
Winner of IPL 2022: Gujarat Titans
```

```
team_wins = df["WinningTeam"].value_counts()
print("Number of matches won by each team in IPL 2022:\n", team_wins)
```

```
Number of matches won by each team in IPL 2022:
 Gujarat Titans                 12
Rajasthan Royals               10
Royal Challengers Bangalore     9







Name: WinningTeam, dtype: int64
```

```
most_common_city = df["City"].value_counts().idxmax()
matches_in_city = df["City"].value_counts().max()
print("City that hosted the maximum number of matches in IPL 2022:",
most_common_city)
print("Number of matches hosted in that city:", matches_in_city)
```

```
City that hosted the maximum number of matches in IPL 2022: Mumbai
Number of matches hosted in that city: 48
```

```
most_umpire1 = df["Umpire1"].value_counts().idxmax()
print("umpiring done in most matches by 1st umpire:", most_umpire1)
most_umpire_count  =  df["Umpire1"].value_counts().max()
print("No.of Times umpiring done in matches by 1st umpire:",
most_umpire_count)
```

```
umpiring done in most matches by 1st umpire: AK Chaudhary
No.of Times umpiring done in most matches by 1st umpire: 11
```

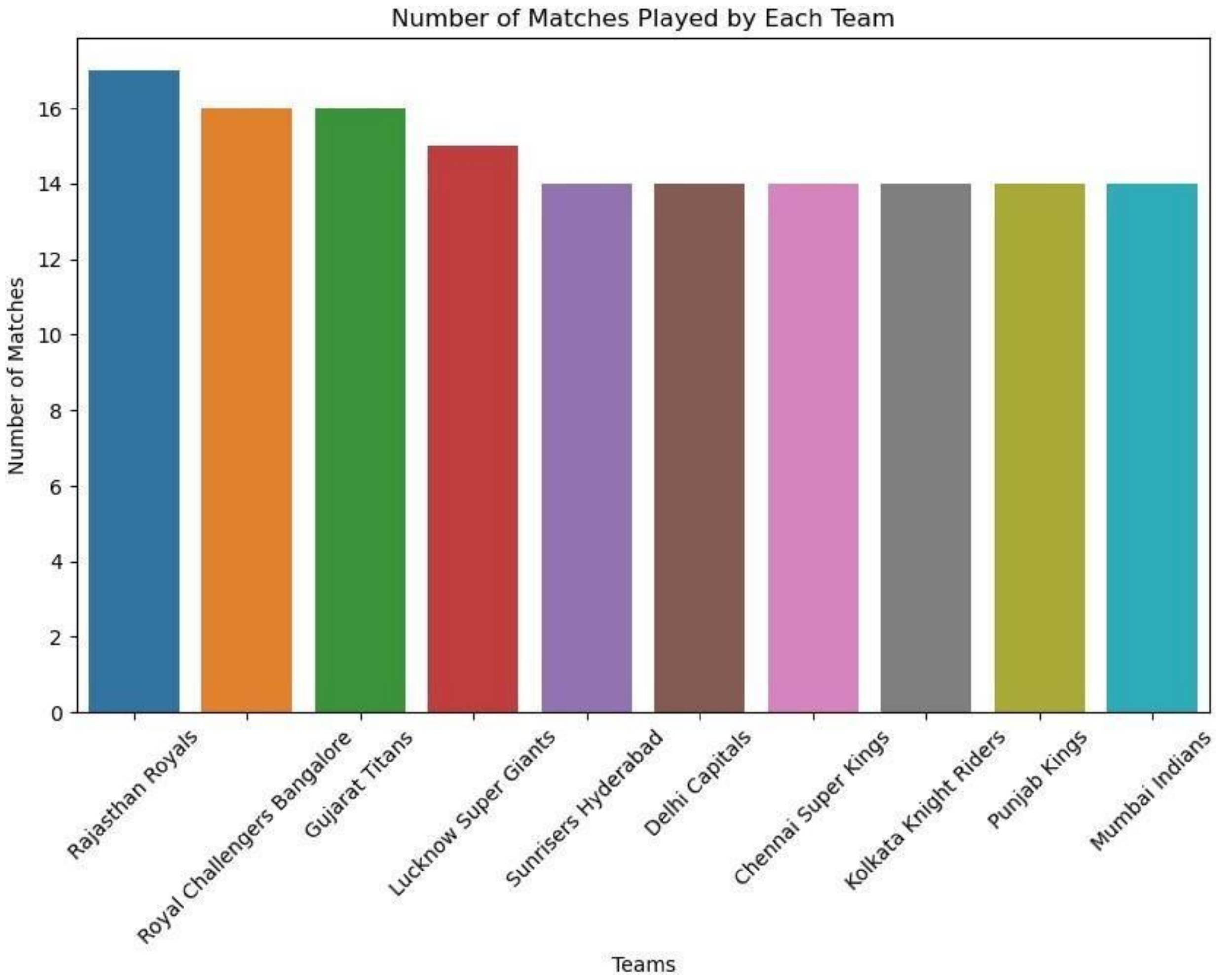
```
most_umpire2 = df["Umpire2"].value_counts().idxmax()
print("umpiring done in most matches by 2st umpire:", most_umpire2)
most_umpire_count  =  df["Umpire2"].value_counts().max()
print("No.of Times umpiring done in matches by 2st umpire:",
most_umpire_count)
```

```
umpiring done in most matches by 2st umpire: VK Sharma
No.of Times umpiring done in matches by 2st umpire: 9
```

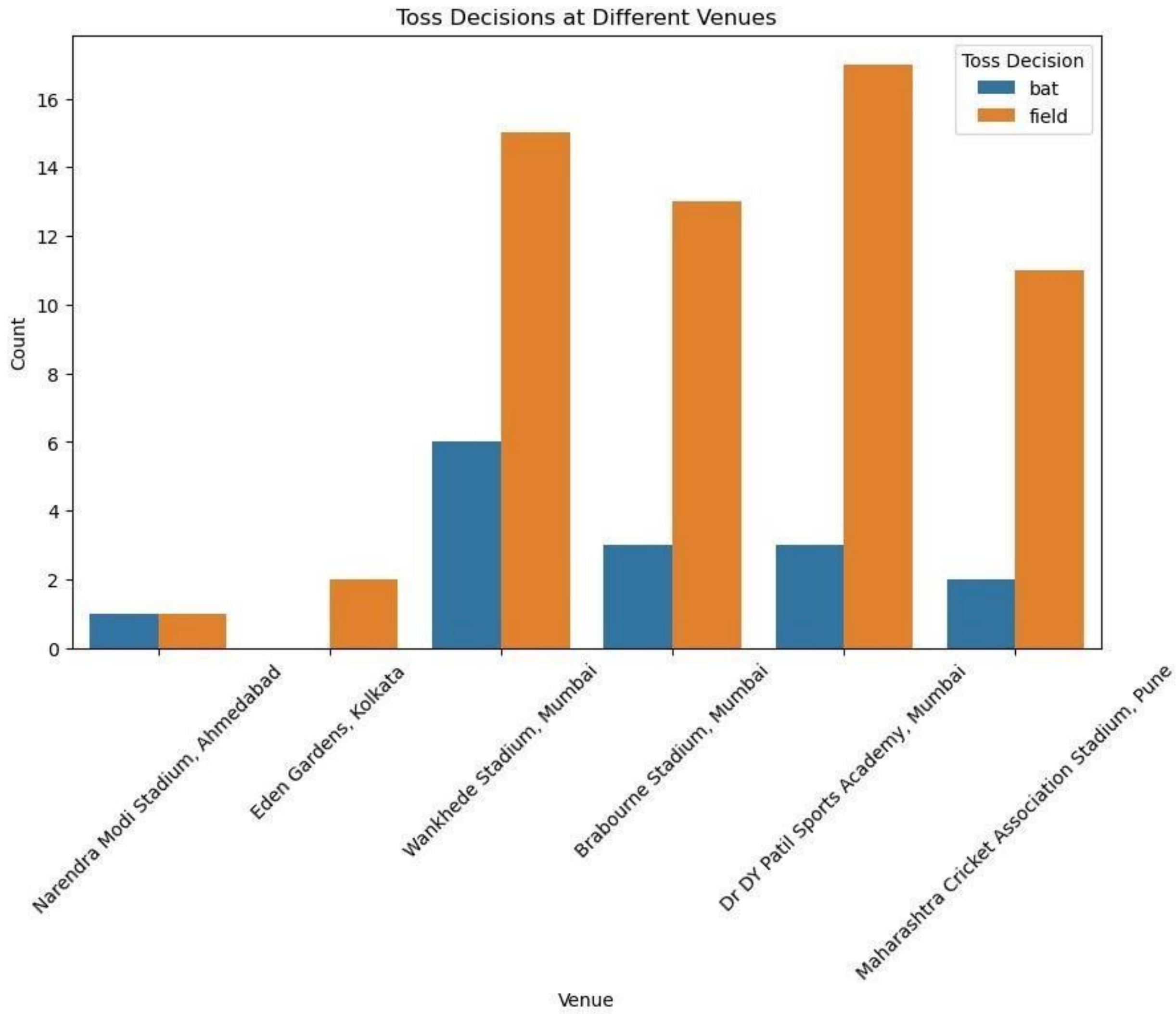
```
player_name = "Kuldeep Yadav"
player_motm_wins = df[df["Player_of_Match"] == player_name].shape[0]
print("Number of matches", player_name, "won the 'Man of the Match'
award:", player_motm_wins)
```

```
Number of matches Kuldeep Yadav won the 'Man of the Match' award: 4
```

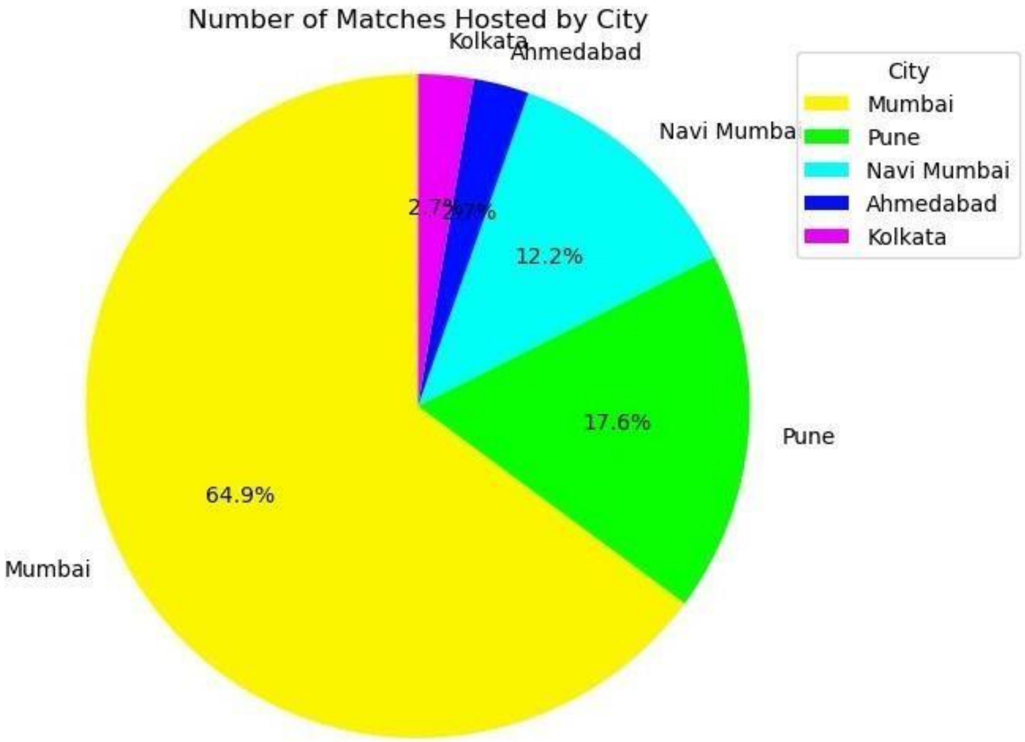
```
team_matches = pd.concat([df['Team1'], df['Team2']]).value_counts()
plt.figure(figsize=(10, 6))
sns.barplot(x=team_matches.index, y=team_matches.values)
plt.title('Number of Matches Played by Each Team')
plt.xlabel('Teams')
plt.ylabel('Number of Matches')
plt.xticks(rotation=45)
plt.show()
```



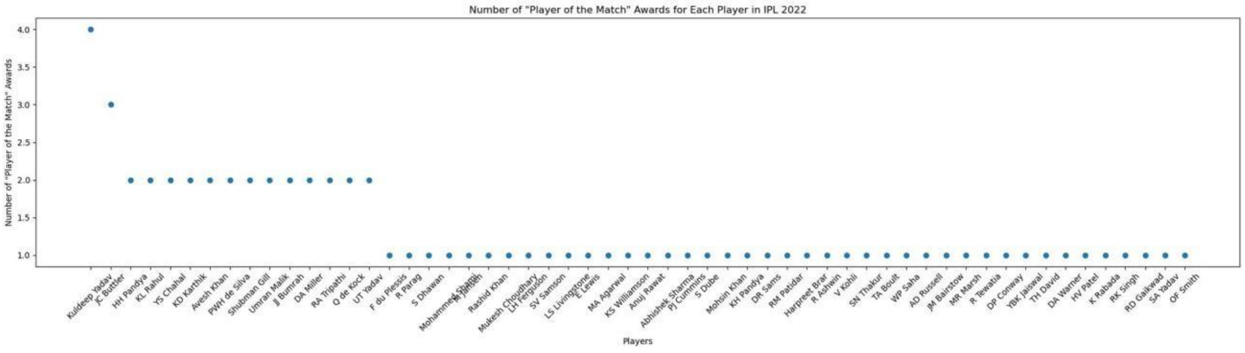
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Venue', hue='TossDecision', data=df)
plt.title('Toss Decisions at Different Venues')
plt.xlabel('Venue')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Toss Decision')
plt.show()
```

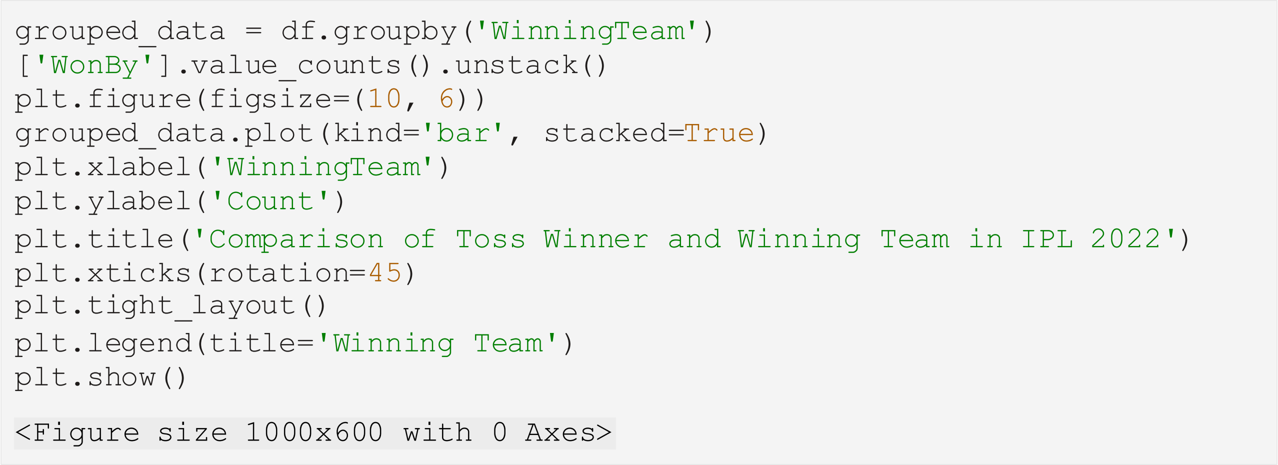


```
city_counts = df['City'].value_counts()
num_cities = len(city_counts)
plt.figure(figsize=(10, 6))
colors = sns.color_palette('hsv', num_cities)
plt.pie(city_counts, labels=city_counts.index, colors=colors,
autopct='%1.1f%%', startangle=90)
plt.title('Number of Matches Hosted by City')
plt.axis('equal')
plt.legend(title='City', loc='best')
plt.show()
```



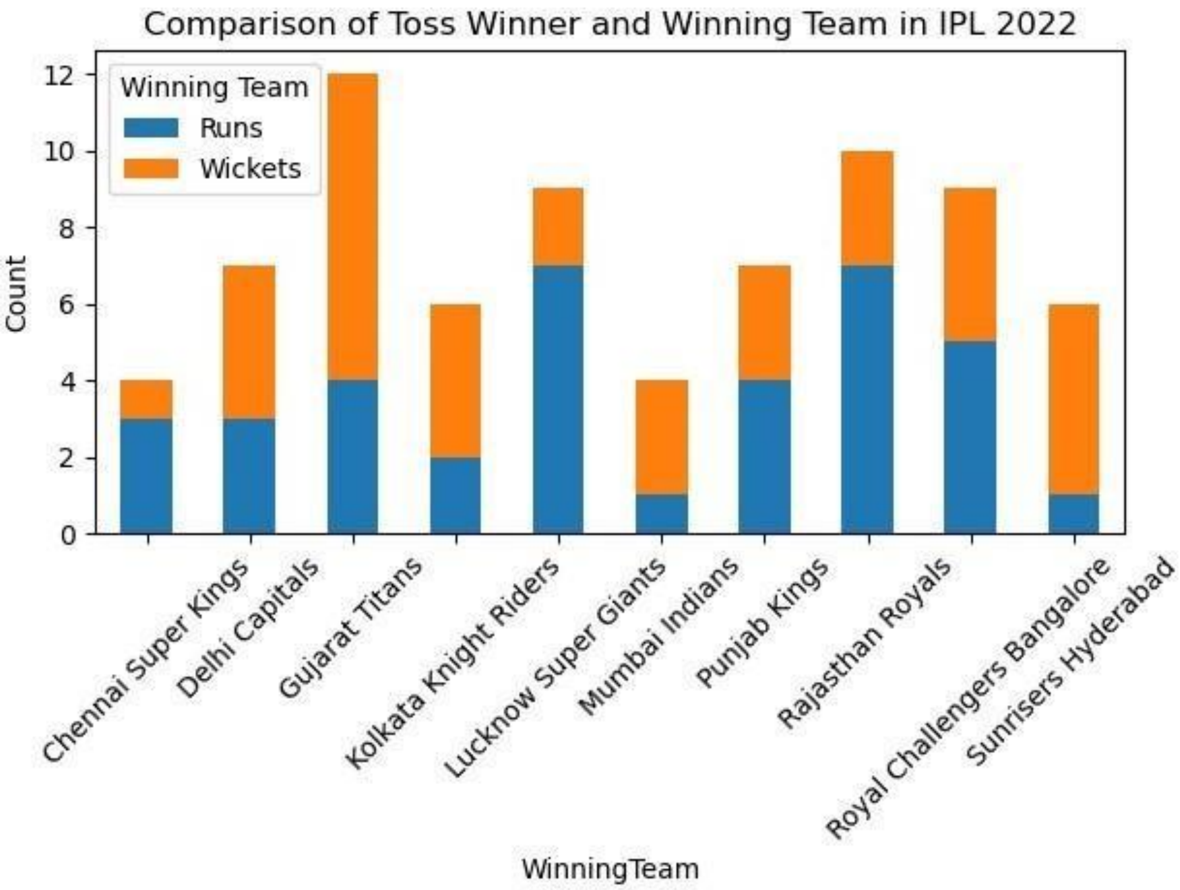
```
pom_counts = df['Player_of_Match'].value_counts()
players = pom_counts.index
awards = pom_counts.values
plt.figure(figsize=(21, 6))
plt.scatter(players, awards)
plt.xlabel('Players')
plt.ylabel('Number of "Player of the Match" Awards')
plt.title('Number of "Player of the Match" Awards for Each Player in
IPL 2022')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

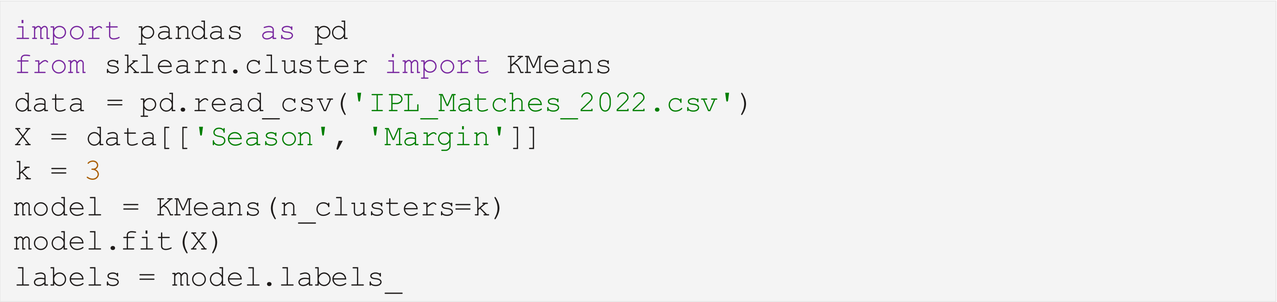




```
grouped_data = df.groupby('WinningTeam')
['WonBy'].value_counts().unstack()
plt.figure(figsize=(10, 6))
grouped_data.plot(kind='bar', stacked=True)
plt.xlabel('WinningTeam')
plt.ylabel('Count')
plt.title('Comparison of Toss Winner and Winning Team in IPL 2022')
plt.xticks(rotation=45)
plt.tight_layout()
plt.legend(title='Winning Team')
plt.show()

<Figure size 1000x600 with 0 Axes>
```





```
import pandas as pd
from sklearn.cluster import KMeans
data = pd.read_csv('IPL_Matches_2022.csv')
X = data[['Season', 'Margin']]
k = 3
model = KMeans(n_clusters=k)
model.fit(X)
labels = model.labels_
```

```python
data['Cluster'] = labels
print('Cluster Centers:')
print(model.cluster_centers_)
```

```
Cluster Centers:
[[2022.           61.4       ]
 [2022.            6.92307692]
 [2022.           23.5       ]]
```

```
/Users/sujitgore/anaconda3/lib/python3.10/site-packages/sklearn/
cluster/_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
```

```python
import pandas as pd
from sklearn.linear_model import LinearRegression
data = pd.read_csv('IPL_Matches_2022.csv')
X = data[['Margin']]
y = data['ID']
model = LinearRegression()
model.fit(X, y)
predictions = model.predict(X)
print('Intercept:', model.intercept_)
print('Coefficient:', model.coef_)
```

```
Intercept: 1304678.118042852
Coefficient: [-9.30074456]
```

THANK YOU!