

```
# Simple Football Game "Merancang Simulasi Permainan Bola Sederhana"
```

```
import math
import random
```

```
#Lambda value in Poisson distribution for higher rated team
```

```
lambOne = 1.148698355
```

```
#Lambda value for lower rated team
```

```
lambTwo = 0.8705505633
```

```
#Poisson distribution calculating goals scored by the home team
```

```
def homeMatch(homeRating,awayRating):
```

```
    global lambOne
```

```
    global x
```

```
    global y
```

```
    if x == y:
```

```
        raise ValueError
```

```
    else:
```

```
        lamb = lambOne**(int(homeRating)-int(awayRating))
```

```
        homeScore = 0
```

```
        z = random.random()
```

```
        while z > 0:
```

```
            z = z - ((lamb**homeScore * math.exp(lamb * -
1)))/(math.factorial(homeScore)))
```

```
            homeScore += 1
```

```
        return (homeScore-1)
```

```
#Poisson distribution calculating goals scored by away team
```

```
def awayMatch(homeRating,awayRating):
```

```
    global lambTwo
```

```
    global x
```

```
    global y
```

```
#This check is to stop a team playing itself
```

```
    if x == y:
```

```
        raise ValueError
```

```
    else:
```

```
        lamb = lambTwo**(int(homeRating)-int(awayRating))
```

```
        awayScore = 0
```

```
        z = random.random()
```

```
        while z > 0:
```

```
            z = z - ((lamb**awayScore * math.exp(lamb * -
1)))/(math.factorial(awayScore)))
```

```
            awayScore += 1
```

```
        return (awayScore-1)
```

```
#Selecting number of teams in league
```

```
leagueSize = int(input("Enter Number of Teams in league: "))
```

```
#Initialising empty lists
```

```
teamNames = []
```

```
teamSkill = []
```

```

teamPoints = []
teamFor = []
teamAgainst = []
teamWins = []
teamDraws = []
teamLosses = []

#Populating lists with number of zeroes equal to the number of teams
(one zero for each)
for x in range(leagueSize):
    teamPoints += [0]
    teamFor += [0]
    teamAgainst += [0]
    teamWins += [0]
    teamDraws += [0]
    teamLosses += [0]

#Entering names and skill ratings for each team
for i in range(leagueSize):
    teamNames += [input("Enter team "+str(i+1)+" name: ")]
for j in range(leagueSize):
    teamSkill += [input("Enter "+teamNames[j]+" skill: ")]

#Initialising variables
homeScore = 0
awayScore = 0

#The season begins - each team plays all of its home games in one go
for x in range(leagueSize):
    #input("Press enter to continue ")
    print("=====")
    print(teamNames[x]+'s home games: ')
    print("=====\\n")
    for y in range(leagueSize):
        error = 0
        try:
            homeScore = homeMatch(teamSkill[x],teamSkill[y])
            #Skipping a game to stop a team playing itself
        except ValueError:
            pass
        error += 1
        try:
            awayScore = awayMatch(teamSkill[x],teamSkill[y])
        except ValueError:
            pass
        if error == 0:
            #Updating lists
            print(teamNames[x],homeScore,"-",awayScore,teamNames[y],"\\
n")
            teamFor[x] += homeScore

```

```

        teamFor[y] += awayScore
        teamAgainst[x] += awayScore
        teamAgainst[y] += homeScore
        if homeScore > awayScore:
            teamWins[x] += 1
            teamLosses[y] += 1
            teamPoints[x] += 3
        elif homeScore == awayScore:
            teamDraws[x] += 1
            teamDraws[y] += 1
            teamPoints[x] += 1
            teamPoints[y] += 1
        else:
            teamWins[y] += 1
            teamLosses[x] += 1
            teamPoints[y] += 3
    else:
        pass

#Printing table (unsorted)
print("Final table: ")
for x in range(leagueSize):
    #Lots of formatting
    print(teamNames[x]+(15-len(teamNames[x]))*" "+" Skill:
"+str(teamSkill[x])+(5-len(str(teamSkill[x]))*" "+" Points:
"+str(teamPoints[x])+(5-len(str(teamPoints[x]))*" "+" For:
"+str(teamFor[x])+(5-len(str(teamFor[x]))*" "+" Against:
"+str(teamAgainst[x])+(5-len(str(teamPoints[x]))*" "+" Goal
difference: "+str(teamFor[x]-teamAgainst[x])+(5-len(str(teamFor[x]-
teamAgainst[x]))*" "+" Wins: "+str(teamWins[x])+(5-
len(str(teamWins[x]))*" "+" Draws: "+str(teamDraws[x])+(5-
len(str(teamDraws[x]))*" "+" Losses: "+str(teamLosses[x])+(5-
len(str(teamLosses[x]))*" ")
    teamPoints.sort()
print(teamPoints)

```

Enter Number of Teams in league: 2

Enter team 1 name: MUN

Enter team 2 name: CFC

Enter MUN skill: 80

Enter CFC skill: 90

=====

MUN's home games:

=====

MUN 0 - 1 CFC

=====

CFC's home games:

=====

CFC 2 - 0 MUN

Final table:

MUN	Skill: 80	Points: 0	For: 0	Against: 3
Goal difference: -3	Wins: 0	Draws: 0	Losses: 2	
CFC	Skill: 90	Points: 6	For: 3	Against: 0
Goal difference: 3	Wins: 2	Draws: 0	Losses: 0	

[0, 6]