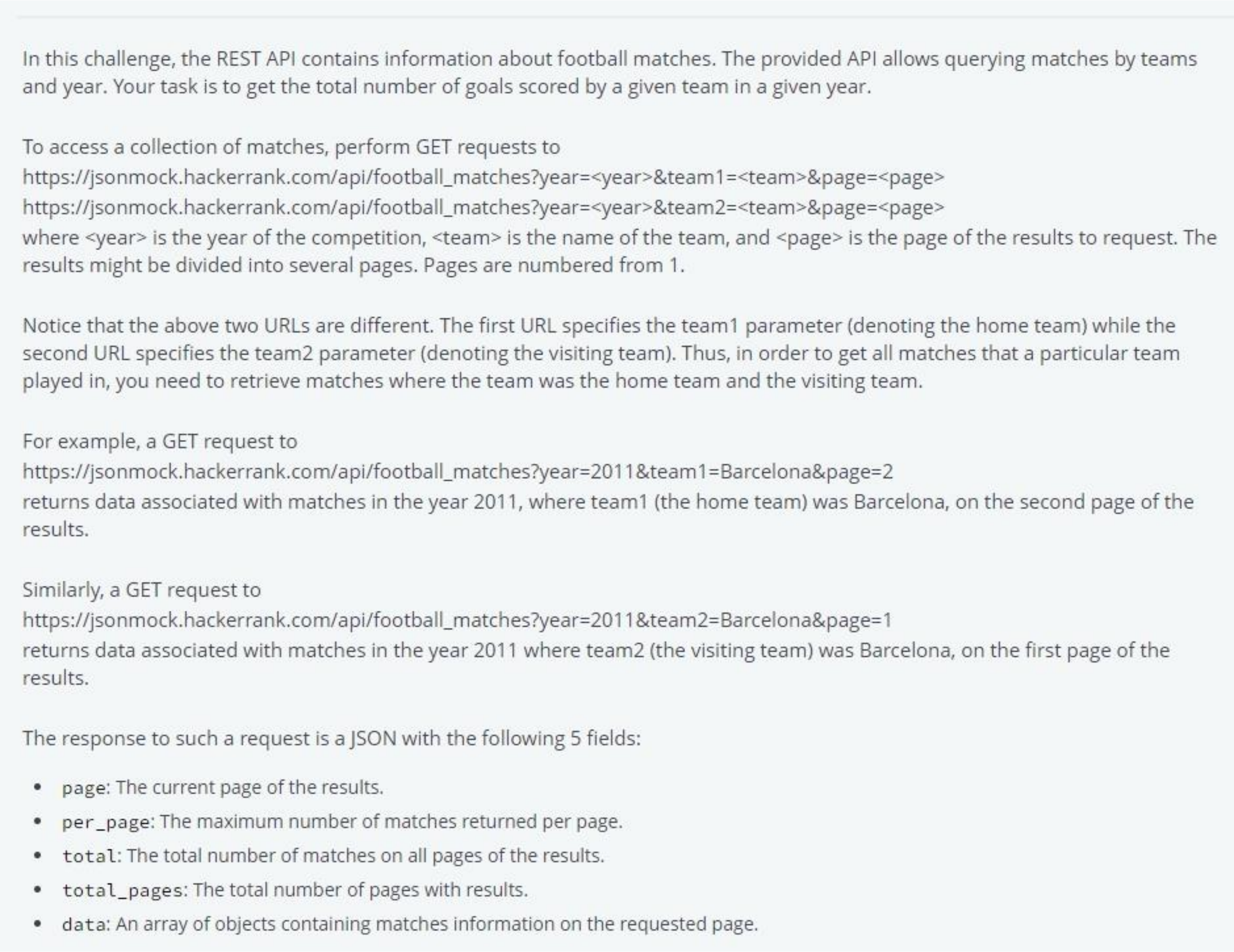
1. REST API: Total Goals by a Team



In this challenge, the REST API contains information about football matches. The provided API allows querying matches by teams and year. Your task is to get the total number of goals scored by a given team in a given year.

To access a collection of matches, perform GET requests to
https://jsonmock.hackerrank.com/api/football_matches?year=<year>&team1=<team>&page=<page>
https://jsonmock.hackerrank.com/api/football_matches?year=<year>&team2=<team>&page=<page>
where <year> is the year of the competition, <team> is the name of the team, and <page> is the page of the results to request. The results might be divided into several pages. Pages are numbered from 1.

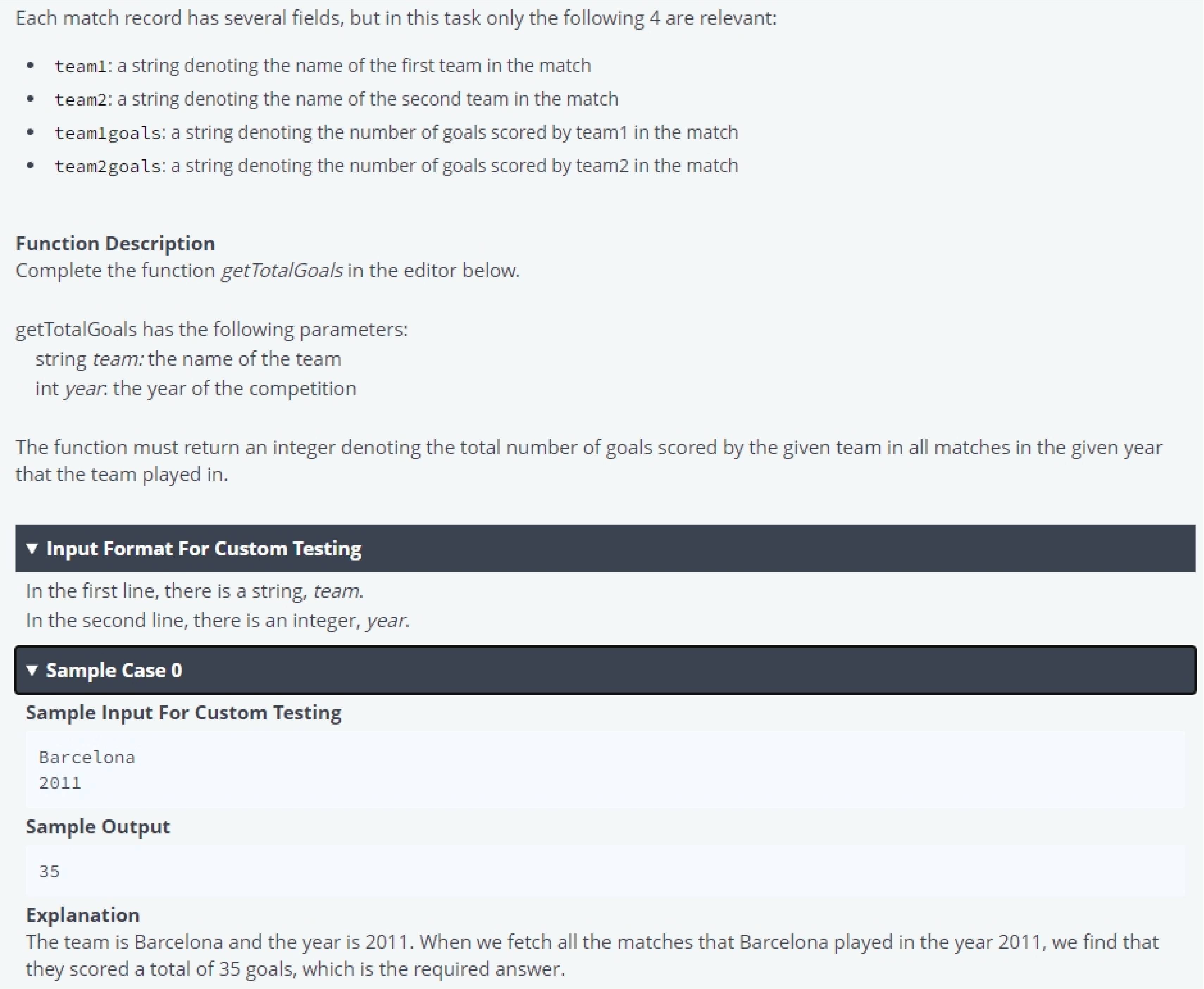
Notice that the above two URLs are different. The first URL specifies the team1 parameter (denoting the home team) while the second URL specifies the team2 parameter (denoting the visiting team). Thus, in order to get all matches that a particular team played in, you need to retrieve matches where the team was the home team and the visiting team.

For example, a GET request to
https://jsonmock.hackerrank.com/api/football_matches?year=2011&team1=Barcelona&page=2
returns data associated with matches in the year 2011, where team1 (the home team) was Barcelona, on the second page of the results.

Similarly, a GET request to
https://jsonmock.hackerrank.com/api/football_matches?year=2011&team2=Barcelona&page=1
returns data associated with matches in the year 2011 where team2 (the visiting team) was Barcelona, on the first page of the results.

The response to such a request is a JSON with the following 5 fields:

- `page`: The current page of the results.
- `per_page`: The maximum number of matches returned per page.
- `total`: The total number of matches on all pages of the results.
- `total_pages`: The total number of pages with results.
- `data`: An array of objects containing matches information on the requested page.



Each match record has several fields, but in this task only the following 4 are relevant:

- `team1`: a string denoting the name of the first team in the match
- `team2`: a string denoting the name of the second team in the match
- `team1goals`: a string denoting the number of goals scored by team1 in the match
- `team2goals`: a string denoting the number of goals scored by team2 in the match

Function Description
Complete the function *getTotalGoals* in the editor below.

getTotalGoals has the following parameters:
string *team:* the name of the team
int *year*: the year of the competition

The function must return an integer denoting the total number of goals scored by the given team in all matches in the given year that the team played in.

▼ Input Format For Custom Testing

In the first line, there is a string, *team*.
In the second line, there is an integer, *year*.

▼ Sample Case 0

Sample Input For Custom Testing

```
Barcelona
2011
```

Sample Output

```
35
```

Explanation
The team is Barcelona and the year is 2011. When we fetch all the matches that Barcelona played in the year 2011, we find that they scored a total of 35 goals, which is the required answer.

Exploring the API data:

```
import requests
import json

team = 'Barcelona'
year = 2011

url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&team1=' +team+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
r
```

```
{'page': '1',
 'per_page': 10,
 'total': 6,
 'total_pages': 1,
 'data': [{'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'Barcelona',
   'team2': 'AC Milan',
   'team1goals': '2',
   'team2goals': '2'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'Barcelona',
   'team2': 'Viktoria Plzen',
   'team1goals': '2',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'Barcelona',
   'team2': 'BATE Borisov',
   'team1goals': '4',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'R16',
   'team1': 'Barcelona',
   'team2': 'Bayer Leverkusen',
   'team1goals': '7',
   'team2goals': '1'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'QF',
   'team1': 'Barcelona',
   'team2': 'AC Milan',
   'team1goals': '3',
   'team2goals': '1'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'SF',
   'team1': 'Barcelona',
   'team2': 'Chelsea',
   'team1goals': '2',
   'team2goals': '2'}]}
```

```
goals = 0
r_data = r['data']
for record in r_data:
    goals += int(record['team1goals'])
goals
20
```

```
url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&team2=' +team+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
r
{'page': '1',
 'per_page': 10,
 'total': 6,
 'total_pages': 1,
 'data': [{'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'BATE Borisov',
   'team2': 'Barcelona',
   'team1goals': '0',
   'team2goals': '5'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'Viktoria Plzen',
   'team2': 'Barcelona',
   'team1goals': '0',
   'team2goals': '4'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupH',
   'team1': 'AC Milan',
   'team2': 'Barcelona',
   'team1goals': '2',
   'team2goals': '3'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'R16',
   'team1': 'Bayer Leverkusen',
   'team2': 'Barcelona',
   'team1goals': '1',
   'team2goals': '3'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'QF',
   'team1': 'AC Milan',
   'team2': 'Barcelona',
   'team1goals': '0',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'SF',
   'team1': 'Chelsea',
   'team2': 'Barcelona',
   'team1goals': '1',
   'team2goals': '0'}]}
```

```
goals = 0
r_data = r['data']
for record in r_data:
    goals += int(record['team2goals'])
goals
```

```
15
```

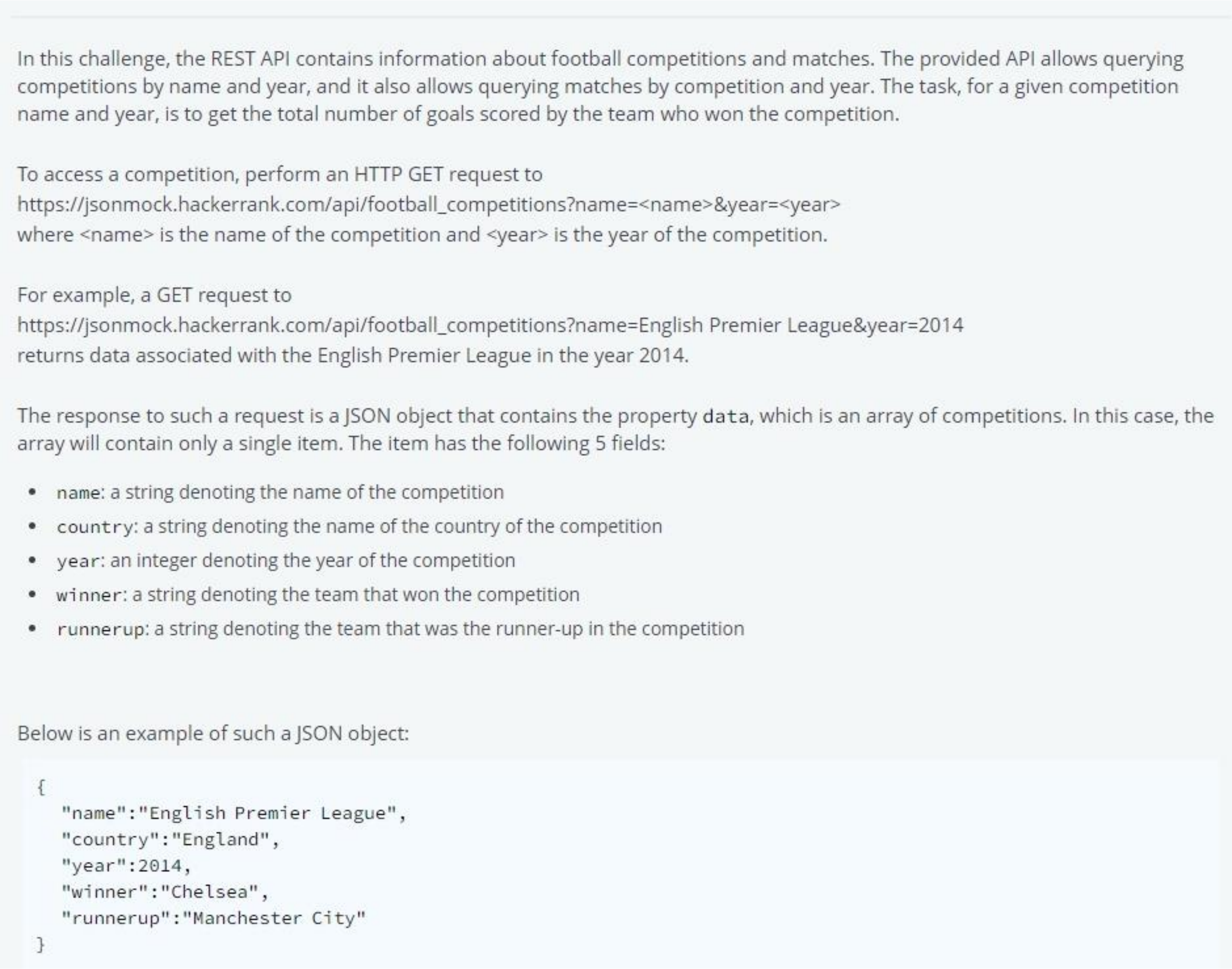
Putting them altogether:

```
import requests
import json

team = 'Barcelona'
year = 2011
goals = 0

for tm in ['team1', 'team2']:
    url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&' +tm+ '=' +team+ '&page=1'
    response = requests.request('GET', url, headers={}, data={})
    total_pages = json.loads(response.text.encode('utf8'))['total_pages']
    for i in range(1, total_pages+1):
        url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&' +tm+ '=' +team+ '&page=' +str(i)
        response = requests.request('GET', url, headers={}, data={})
        r = json.loads(response.text.encode('utf8'))
        r_data = r['data']
        for record in r_data:
            goals += int(record[tm+'goals'])

goals
```

```
35
```

Compiled successfully. All available test cases passed

2. REST API: Football Competition Winner's Goals



In this challenge, the REST API contains information about football competitions and matches. The provided API allows querying competitions by name and year, and it also allows querying matches by competition and year. The task, for a given competition name and year, is to get the total number of goals scored by the team who won the competition.

To access a competition, perform an HTTP GET request to
https://jsonmock.hackerrank.com/api/football_competitions?name=<name>&year=<year>
where <name> is the name of the competition and <year> is the year of the competition.

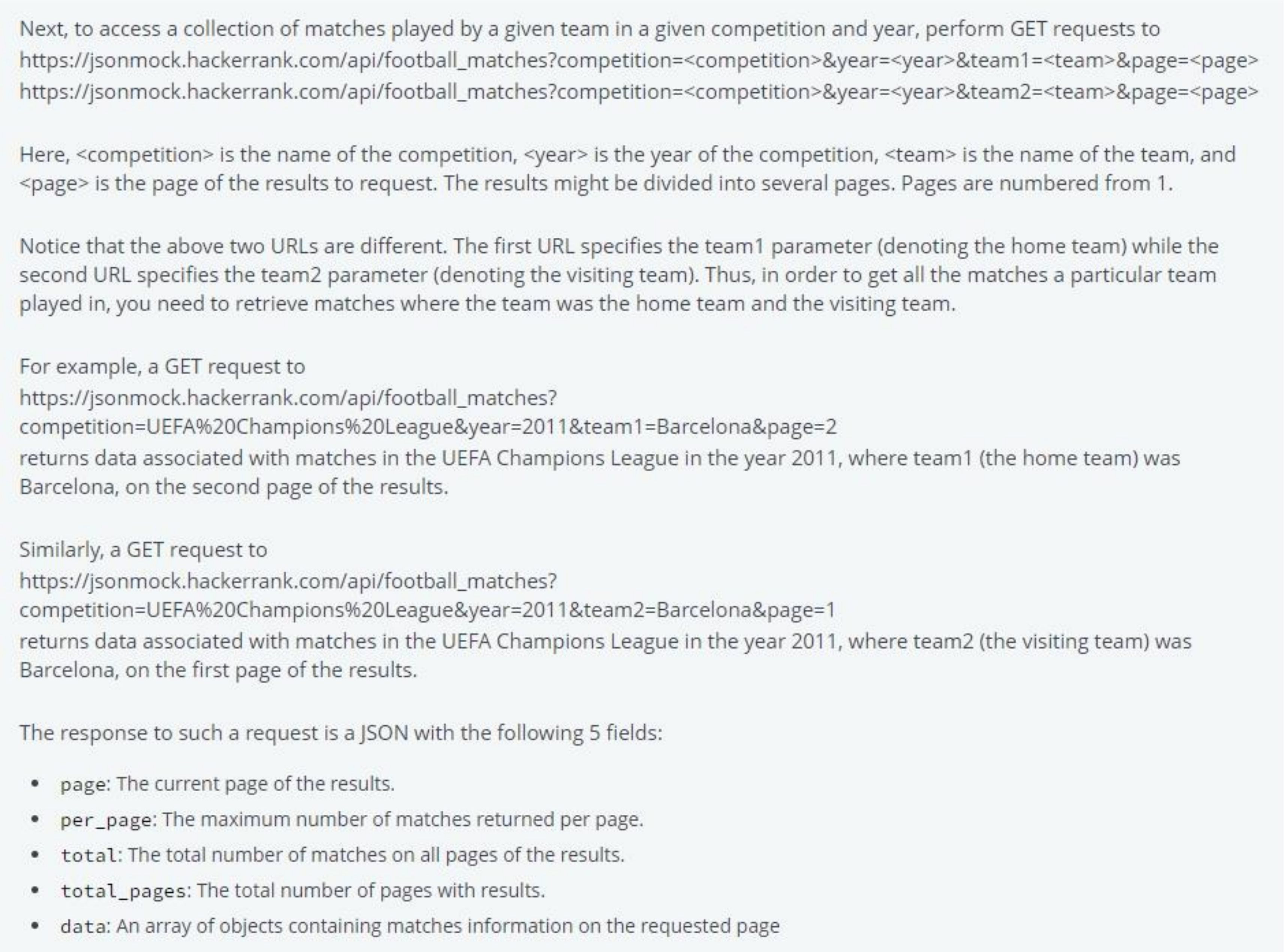
For example, a GET request to
https://jsonmock.hackerrank.com/api/football_competitions?name=English Premier League&year=2014
returns data associated with the English Premier League in the year 2014.

The response to such a request is a JSON object that contains the property `data`, which is an array of competitions. In this case, the array will contain only a single item. The item has the following 5 fields:

- `name`: a string denoting the name of the competition
- `country`: a string denoting the name of the country of the competition
- `year`: an integer denoting the year of the competition
- `winner`: a string denoting the team that won the competition
- `runnerup`: a string denoting the team that was the runner-up in the competition

Below is an example of such a JSON object:

```
{
  "name":"English Premier League",
  "country":"England",
  "year":2014,
  "winner":"Chelsea",
  "runnerup":"Manchester City"
}
```



Next, to access a collection of matches played by a given team in a given competition and year, perform GET requests to
https://jsonmock.hackerrank.com/api/football_matches?competition=<competition>&year=<year>&team1=<team>&page=<page>
https://jsonmock.hackerrank.com/api/football_matches?competition=<competition>&year=<year>&team2=<team>&page=<page>

Here, <competition> is the name of the competition, <year> is the year of the competition, <team> is the name of the team, and <page> is the page of the results to request. The results might be divided into several pages. Pages are numbered from 1.

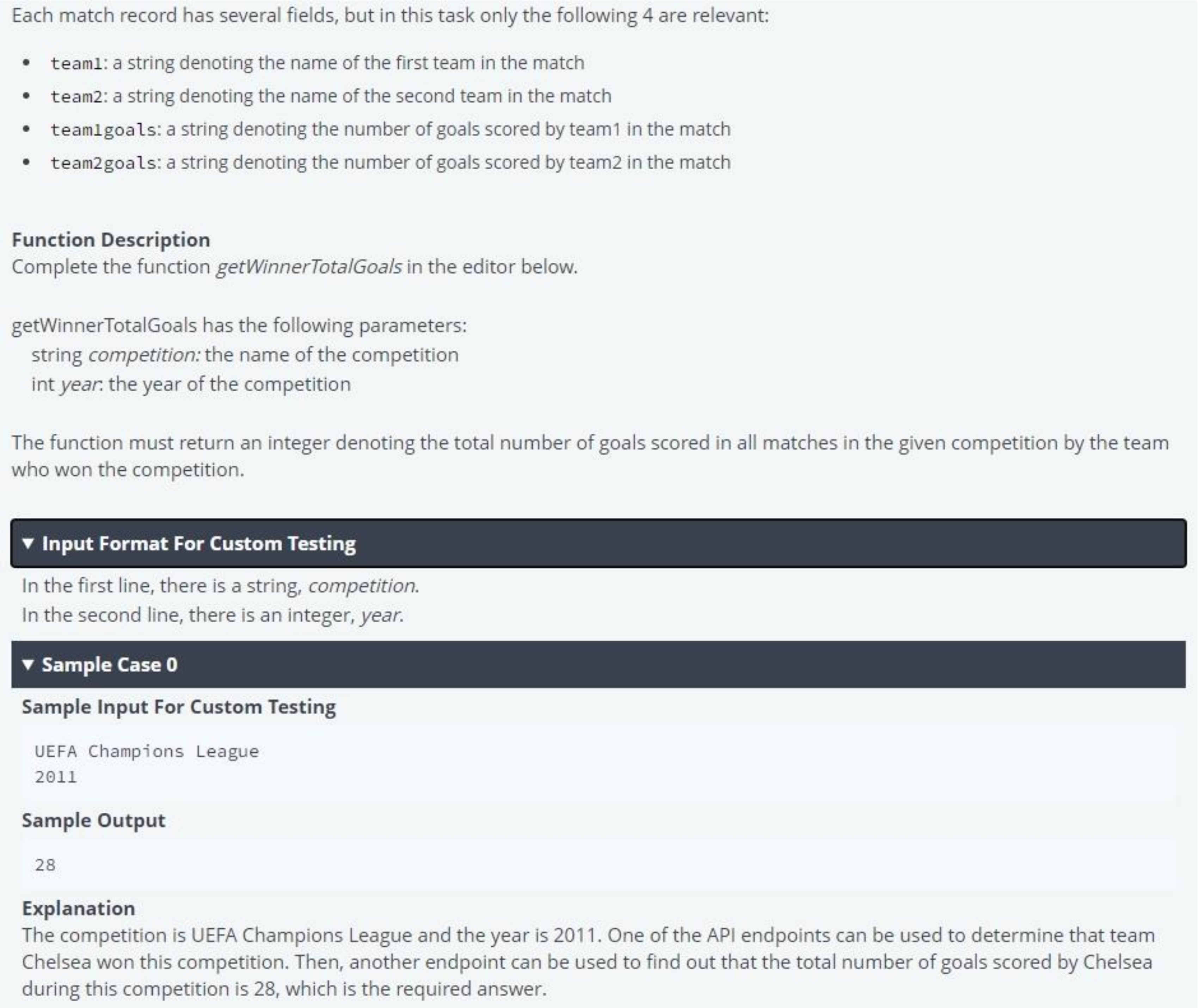
Notice that the above two URLs are different. The first URL specifies the team1 parameter (denoting the home team) while the second URL specifies the team2 parameter (denoting the visiting team). Thus, in order to get all the matches a particular team played in, you need to retrieve matches where the team was the home team and the visiting team.

For example, a GET request to
https://jsonmock.hackerrank.com/api/football_matches?competition=UEFA%20Champions%20League&year=2011&team1=Barcelona&page=2
returns data associated with matches in the UEFA Champions League in the year 2011, where team1 (the home team) was Barcelona, on the second page of the results.

Similarly, a GET request to
https://jsonmock.hackerrank.com/api/football_matches?competition=UEFA%20Champions%20League&year=2011&team2=Barcelona&page=1
returns data associated with matches in the UEFA Champions League in the year 2011, where team2 (the visiting team) was Barcelona, on the first page of the results.

The response to such a request is a JSON with the following 5 fields:

- `page`: The current page of the results.
- `per_page`: The maximum number of matches returned per page.
- `total`: The total number of matches on all pages of the results.
- `total_pages`: The total number of pages with results.
- `data`: An array of objects containing matches information on the requested page



Each match record has several fields, but in this task only the following 4 are relevant:

- `team1`: a string denoting the name of the first team in the match
- `team2`: a string denoting the name of the second team in the match
- `team1goals`: a string denoting the number of goals scored by team1 in the match
- `team2goals`: a string denoting the number of goals scored by team2 in the match

Function Description
Complete the function *getWinnerTotalGoals* in the editor below.

getWinnerTotalGoals has the following parameters:
string *competition:* the name of the competition
int *year*: the year of the competition

The function must return an integer denoting the total number of goals scored in all matches in the given competition by the team who won the competition.

▼ Input Format For Custom Testing

In the first line, there is a string, *competition*.
In the second line, there is an integer, *year*.

▼ Sample Case 0

Sample Input For Custom Testing

```
UEFA Champions League
2011
```

Sample Output

```
28
```

Explanation
The competition is UEFA Champions League and the year is 2011. One of the API endpoints can be used to determine that team Chelsea won this competition. Then, another endpoint can be used to find out that the total number of goals scored by Chelsea during this competition is 28, which is the required answer.

Exploring the API data:

```
import requests
import json

competition = 'UEFA Champions League'
year = 2011

url = 'https://jsonmock.hackerrank.com/api/football_competitions?name=' +competition+ '&year=' +str(year)
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
r
```

```
{'page': 1,
 'per_page': 10,
 'total': 1,
 'total_pages': 1,
 'data': [{'name': 'UEFA Champions League',
   'country': '',
   'year': 2011,
   'winner': 'Chelsea',
   'runnerup': 'Bayern Munich'}]}
```

```
winner = 'Chelsea'

url = 'https://jsonmock.hackerrank.com/api/football_matches?competition=' +competition+ '&year=' +str(year)+ '&team1=' +winner+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
r
```

```
{'page': '1',
 'per_page': 10,
 'total': 7,
 'total_pages': 1,
 'data': [{'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupE',
   'team1': 'Chelsea',
   'team2': 'Bayer Leverkusen',
   'team1goals': '2',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupE',
   'team1': 'Chelsea',
   'team2': 'KRC Genk',
   'team1goals': '5',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'GroupE',
   'team1': 'Chelsea',
   'team2': 'Valencia CF',
   'team1goals': '3',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'R16',
   'team1': 'Chelsea',
   'team2': 'SSC Napoli',
   'team1goals': '3',
   'team2goals': '1'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'QF',
   'team1': 'Chelsea',
   'team2': 'SL Benfica',
   'team1goals': '2',
   'team2goals': '1'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'SF',
   'team1': 'Chelsea',
   'team2': 'Barcelona',
   'team1goals': '1',
   'team2goals': '0'},
  {'competition': 'UEFA Champions League',
   'year': 2011,
   'round': 'final',
   'team1': 'Chelsea',
   'team2': 'Bayern Munich',
   'team1goals': '5',
   'team2goals': '4'}]}
```

```
goals = 0
r_data = r['data']
for record in r_data:
    goals += int(record['team1goals'])
goals
```

```
21
```

```
url = 'https://jsonmock.hackerrank.com/api/football_matches?competition=' +competition+ '&year=' +str(year)+ '&team2=' +winner+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
r
```

```
{'page': '1',
 'per_page': 10,
 'total': 6,
 'total_pages': 1,
 'data': [{'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'GroupE',
  'team1': 'Valencia CF',
  'team2': 'Chelsea',
  'team1goals': '1',
  'team2goals': '1'},
 {'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'GroupE',
  'team1': 'KRC Genk',
  'team2': 'Chelsea',
  'team1goals': '1',
  'team2goals': '1'},
 {'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'GroupE',
  'team1': 'Bayer Leverkusen',
  'team2': 'Chelsea',
  'team1goals': '2',
  'team2goals': '1'},
 {'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'R16',
  'team1': 'SSC Napoli',
  'team2': 'Chelsea',
  'team1goals': '3',
  'team2goals': '1'},
 {'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'QF',
  'team1': 'SL Benfica',
  'team2': 'Chelsea',
  'team1goals': '0',
  'team2goals': '1'},
 {'competition': 'UEFA Champions League',
  'year': 2011,
  'round': 'SF',
  'team1': 'Barcelona',
  'team2': 'Chelsea',
  'team1goals': '2',
  'team2goals': '2'}]}
```

```
goals = 0
r_data = r['data']
for record in r_data:
    goals += int(record['team2goals'])
goals
```

```
7
```

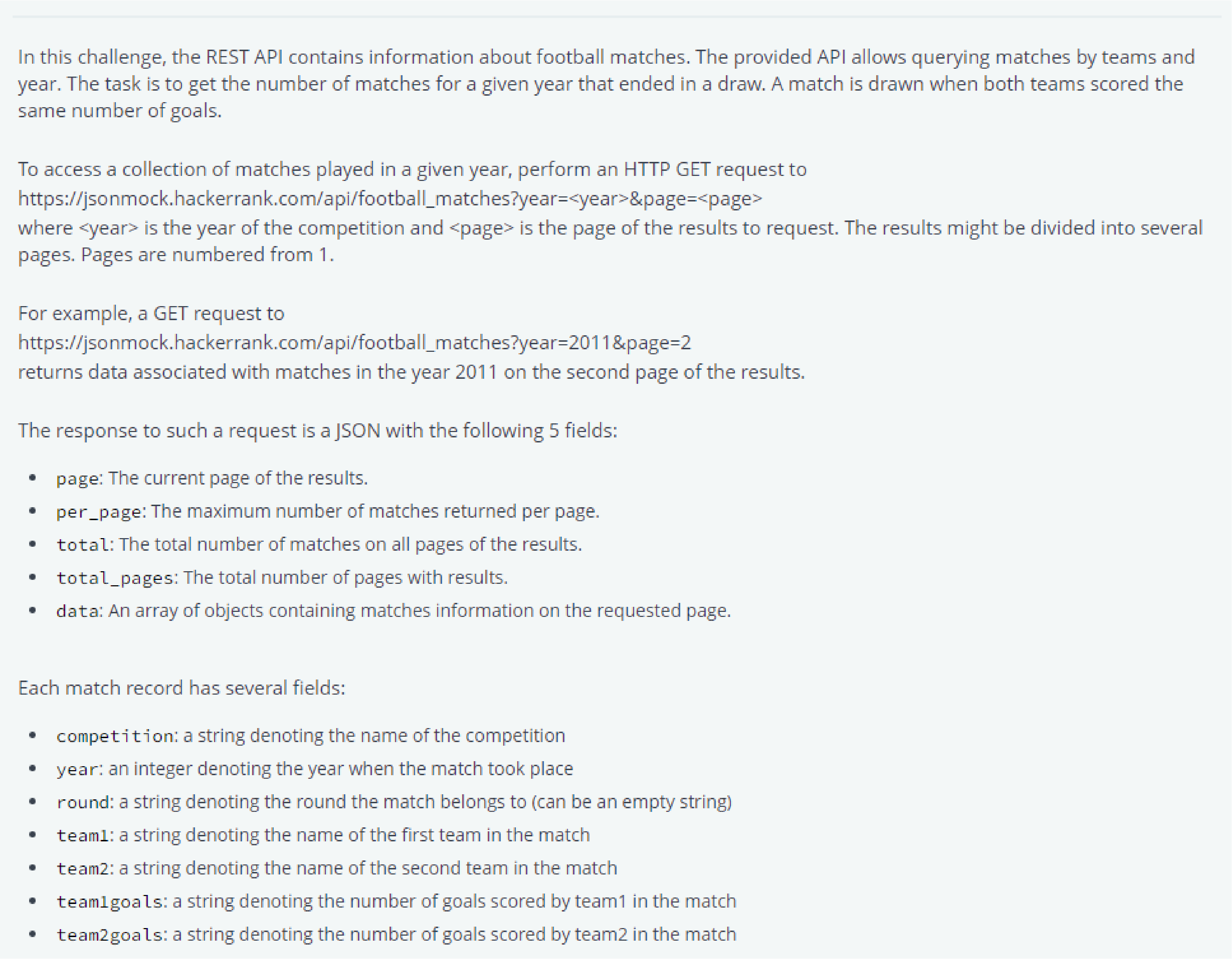
Putting them altogether:

```
import requests
import json

def getWinnerTotalGoals(competition, year):
    goals = 0
    url = 'https://jsonmock.hackerrank.com/api/football_competitions?name=' +competition+ '&year=' +str(year)
    response = requests.request('GET', url, headers={}, data={})
    winner = json.loads(response.text.encode('utf8'))['data'][0]['winner']

    for team in ['team1', 'team2']:
        url = 'https://jsonmock.hackerrank.com/api/football_matches?competition=' +competition+ '&year=' +str(year)+ '&' +team+ '=' +winner+ '&page=1'
        response = requests.request('GET', url, headers={}, data={})
        total_pages = json.loads(response.text.encode('utf8'))['total_pages']
        for i in range(1, total_pages+1):
            url = 'https://jsonmock.hackerrank.com/api/football_matches?competition=' +competition+ '&year=' +str(year)+ '&' +team+ '=' +winner+ '&page=' +str(i)
            response = requests.request('GET', url, headers={}, data={})
            r = json.loads(response.text.encode('utf8'))
            r_data = r['data']
            for record in r_data:
                goals += int(record[team+'goals'])

    return goals

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
    competition = input()
    year = int(input().strip())
    result = getWinnerTotalGoals(competition, year)
    fptr.write(str(result) + '\n')
    fptr.close()
```

Input: competition = 'UEFA Champions League' year = 2011 Output: 28	Input (stdin) English Premier League 2014 Expected Output 73	Input (stdin) La Liga 2012 Expected Output 115

Compiled successfully. All available test cases passed

3. REST API: Number of Drawn Matches



In this challenge, the REST API contains information about football matches. The provided API allows querying matches by teams and year. The task is to get the number of matches for a given year that ended in a draw. A match is drawn when both teams scored the same number of goals.

To access a collection of matches played in a given year, perform an HTTP GET request to
https://jsonmock.hackerrank.com/api/football_matches?year=<year>&page=<page>
where <year> is the year of the competition and <page> is the page of the results to request. The results might be divided into several pages. Pages are numbered from 1.

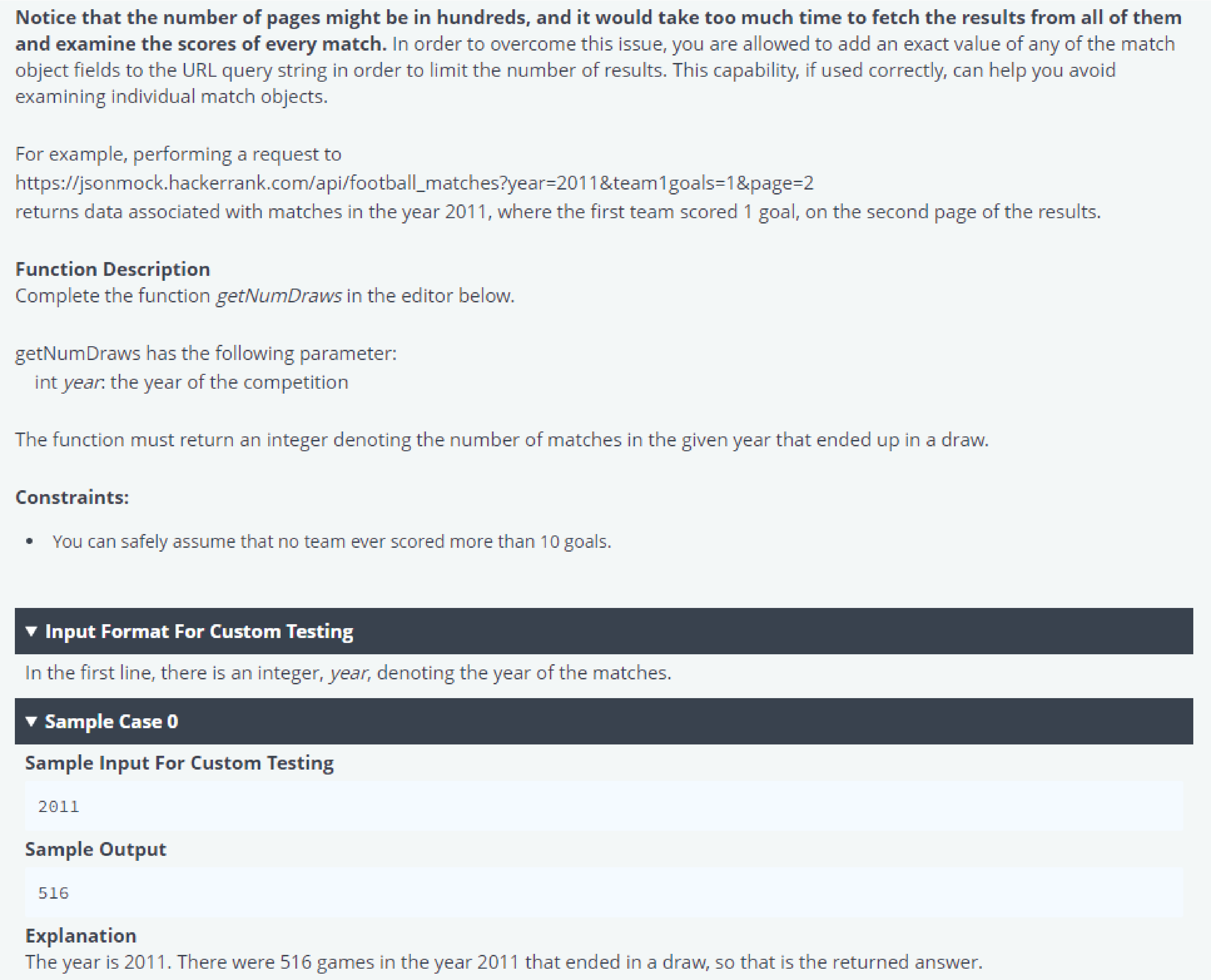
For example, a GET request to
https://jsonmock.hackerrank.com/api/football_matches?year=2011&page=2
returns data associated with matches in the year 2011 on the second page of the results.

The response to such a request is a JSON with the following 5 fields:

- `page`: The current page of the results.
- `per_page`: The maximum number of matches returned per page.
- `total`: The total number of matches on all pages of the results.
- `total_pages`: The total number of pages with results.
- `data`: An array of objects containing matches information on the requested page.

Each match record has several fields:

- `competition`: a string denoting the name of the competition
- `year`: an integer denoting the year when the match took place
- `round`: a string denoting the round the match belongs to (can be an empty string)
- `team1`: a string denoting the name of the first team in the match
- `team2`: a string denoting the name of the second team in the match
- `team1goals`: a string denoting the number of goals scored by team1 in the match
- `team2goals`: a string denoting the number of goals scored by team2 in the match



Notice that the number of pages might be in hundreds, and it would take too much time to fetch the results from all of them and examine the scores of every match. In order to overcome this issue, you are allowed to add an exact value of any of the match object fields to the URL query string in order to limit the number of results. This capability, if used correctly, can help you avoid examining individual match objects.

For example, performing a request to
https://jsonmock.hackerrank.com/api/football_matches?year=2011&team1goals=1&page=2
returns data associated with matches in the year 2011, where the first team scored 1 goal, on the second page of the results.

Function Description
Complete the function *getNumDraws* in the editor below.

getNumDraws has the following parameter:
int *year*: the year of the competition

The function must return an integer denoting the number of matches in the given year that ended up in a draw.

Constraints:

- You can safely assume that no team ever scored more than 10 goals.

▼ **Input Format For Custom Testing**

In the first line, there is an integer, *year*, denoting the year of the matches.

▼ **Sample Case 0**

Sample Input For Custom Testing

```
2011
```

Sample Output

```
516
```

Explanation
The year is 2011. There were 516 games in the year 2011 that ended in a draw, so that is the returned answer.

Exploring the API data:

```
import requests
import json

year = 2011
url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
print(r)
```

```
{'page': '1', 'per_page': 10, 'total': 1951, 'total_pages': 196, 'data': [{'competition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupH', 'team1
': 'Barcelona', 'team2': 'AC Milan', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupG', 'team
1': 'APOEL Nikosia', 'team2': 'Zenit St. Petersburg', 'team1goals': '2', 'team2goals': '1'}, {'competition': 'UEFA Champions League', 'year': 2011, 'round
': 'GroupF', 'team1': 'Borussia Dortmund', 'team2': 'Arsenal', 'team1goals': '1', 'team2goals': '1'}, {'competition': 'UEFA Champions League', 'year': 201
1, 'round': 'GroupH', 'team1': 'Viktoria Plzen', 'team2': 'BATE Borisov', 'team1goals': '1', 'team2goals': '1'}, {'competition': 'UEFA Champions League', '
year': 2011, 'round': 'GroupE', 'team1': 'Chelsea', 'team2': 'Bayer Leverkusen', 'team1goals': '2', 'team2goals': '0'}, {'competition': 'UEFA Champions Lea
gue', 'year': 2011, 'round': 'GroupG', 'team1': 'FC Porto', 'team2': 'Shakhtar Donetsk', 'team1goals': '2', 'team2goals': '1'}, {'competition': 'UEFA Champ
ions League', 'year': 2011, 'round': 'GroupE', 'team1': 'KRC Genk', 'team2': 'Valencia CF', 'team1goals': '0', 'team2goals': '0'}, {'competition': 'UEFA Ch
ampions League', 'year': 2011, 'round': 'GroupF', 'team1': 'Olympiacos', 'team2': 'Olympique Marseille', 'team1goals': '0', 'team2goals': '1'}, {'competiti
on': 'UEFA Champions League', 'year': 2011, 'round': 'GroupD', 'team1': 'AFC Ajax', 'team2': 'Olympique Lyon', 'team1goals': '0', 'team2goals': '0'}, {'com
petition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupC', 'team1': 'Basel', 'team2': 'Otelul Galati', 'team1goals': '2', 'team2goals': '1'}]}
```

There are 1,951 matches in the year 2011 (in 196 pages), too much to iterate through to look for draw matches.

Try to add exact values of team1goals & team2goals so that the match is a draw:

```
year = 2011
g = 2
# url = https://jsonmock.hackerrank.com/api/football_matches?year=2011&team1goals=2&team2goals=2&page=1
url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&team1goals=' +str(g)+ '&team2goals=' +str(g)+ '&page=1'
response = requests.request('GET', url, headers={}, data={})
r = json.loads(response.text.encode('utf8'))
print(r)
```

```
{'page': '1', 'per_page': 10, 'total': 99, 'total_pages': 10, 'data': [{'competition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupH', 'team1': '
Barcelona', 'team2': 'AC Milan', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupB', 'team1':
'Lille OSC', 'team2': 'CSKA Moskva', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', 'year': 2011, 'round': 'GroupG', 'team
1': 'Shakhtar Donetsk', 'team2': 'Zenit St. Petersburg', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', 'year': 2011, 'rou
nd': 'GroupC', 'team1': 'Manchester United', 'team2': 'SL Benfica', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', 'year':
 2011, 'round': 'GroupH', 'team1': 'Viktoria Plzen', 'team2': 'AC Milan', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'UEFA Champions League', '
year': 2011, 'round': 'SF', 'team1': 'Barcelona', 'team2': 'Chelsea', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'English Premier League', 'yea
r': 2011, 'round': '', 'team1': 'Aston Villa', 'team2': 'Queens Park Rangers', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'English Premier Leag
ue', 'year': 2011, 'round': '', 'team1': 'Bolton Wanderers', 'team2': 'West Bromwich Albion', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'Engli
sh Premier League', 'year': 2011, 'round': '', 'team1': 'Everton', 'team2': 'Aston Villa', 'team1goals': '2', 'team2goals': '2'}, {'competition': 'English
Premier League', 'year': 2011, 'round': '', 'team1': 'Fulham', 'team2': 'Manchester City', 'team1goals': '2', 'team2goals': '2'}]}
```

There are 99 matches having results “2-2” draw.

So we can count the number of matches with results “0-0”, “1-1”, “2-2”… … up to “10-10”.

Putting them altogether:

```
import requests
import json

def getNumDraws(year):
    total = 0
    for g in range(0, 11):
        url = 'https://jsonmock.hackerrank.com/api/football_matches?year=' +str(year)+ '&team1goals=' +str(g)+ '&team2goals=' +str(g)+ '&page=1'
        response = requests.request('GET', url, headers={}, data={})
        r = json.loads(response.text.encode('utf8'))
        total += r['total']

    return total

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')
    year = int(input().strip())
    result = getNumDraws(year)
    fptr.write(str(result) + '\n')
    fptr.close()
```

Input:
year = 2011

Output:
516

Compiled successfully. All available test cases passed