

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.

2. 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.