

## 5. REST API: Country Populations

Use the *HTTP GET* method to retrieve information from a database of countries. Query *https://jsonmock.hackerrank.com/api/countries/search?name=s* where *s* is the value of *s*, to find all of the countries that have the substring in their names (not case sensitive). The query response is paginated and can be further accessed by appending to the query string *&page=num* where *num* is the page number.

The query response from the website is a JSON response with the following fields:

*page*: current page number

*per\_page*: maximum number of records to display per page

*total*: total records matching the query

*total\_pages*: the number of pages required to display the matching records

*data*: a json array with the following fields

- *altSpellings*: Alternative spellings of the country's name
- *area*: The country's area (in km<sup>2</sup>)
- *capital*: The name of the country's capital
- *currencies*: The country's official currency
- *latlng*: The capital's latitude and longitude
- *name*: The country's name
- *population*: The country's estimated population
- *timezones*: The timezones that the country spans across
- *topLevelDomain*: The country's top-level domain extension
- *callingCodes*: The international phone calling codes of the country
- *region*: The country's region within a continent
- *subRegion*: The country's sub-region within a continent
- *nativeName*: The country's native or ancient name
- *languages*: A list of languages spoken in the country

Given a substring  $s$ , and the minimum population  $p$ , find the number of country names that contain  $s$  and that have populations greater than  $p$ .

Note: the search is NOT case sensitive

### Function Description

Complete the function *getCountries* in the editor below.

*getCountries* has the following parameter(s):

*string s*: the substring to search for

*int p*: the lower population limit

Returns:

*int*: the number of countries that meet the criteria

### Constraints

- $1 \leq p \leq 10^7$

#### ► Input Format For Custom Testing

#### ▼ Sample Case 0

##### Sample Input

STDIN		Function
-----		-----
un	→	s = 'un'
100090	→	p = 100090

## ▼ Sample Case 0

### Sample Input

STDIN	Function
-----	-----
un	→ s = 'un'
100090	→ p = 100090

### Sample Output

8

### Explanation

Given  $s = 'un'$ , the query is

<https://jsonmock.hackerrank.com/api/countries/search?name=un> and the response is:

```
[
  {
    "name": "United States Minor Outlying Islands",
    "topLevelDomain": [
      ".us"
    ],
    "alpha2Code": "UM",
    "alpha3Code": "UMI",
    "callingCodes": [
      ""
    ],
    "capital": "",
    "altSpellings": [
      "UM"
    ],
    "relevance": "0",
    "region": "Americas",
    "subregion": "Northern America".
```

```

        "UTC+12:00"
    ],
    "borders": [

    ],
    "nativeName": "United States Minor Outlying Islands",
    "numericCode": "581",
    "currencies": [
        "USD"
    ],
    "languages": [
        "en"
    ]
}....

```

The response has 8 countries with populations greater than  $p = 100090$ .

```

#!/bin/python3

import sys
import os
import urllib.request
import json

# Complete the function below.
# https://jsonmock.hackerrank.com/api/countries/search?name=
import requests
def getCountries(s, p):
    # get data and process data from the particular website
    response =
requests.get("https://jsonmock.hackerrank.com/api/countries/search?name=" + s)
    countrys = json.loads(response.content.decode())
    # get the number of total page
    pages = countrys["total_pages"]
    res = 0
    # calculate the countries for every page
    for m in range(1, pages + 1):
        response =
requests.get("https://jsonmock.hackerrank.com/api/countries/search?name=" +
s+"&page=" + str(m))
        countrys = json.loads(response.content.decode())
        countrys = countrys["data"]
        for i in countrys:
            if i["population"] > p:
                res += 1
    return res
f = open(os.environ['OUTPUT_PATH'], 'w')

try:
    _s = str(input())
except:
    _s = None

```

```
_p = int(input());  
  
res = getCountries(_s, _p)  
f.write(str(res) + "\n")  
  
f.close()
```