



Question - 1

FizzBuzz

SCORE: 50 points

Easy

General Programming

Basic Programming

Loops

Conditionals

Core Skills

Problem Solving

In this challenge, consider a range of positive integers from 1 to a limit. For each value, print either a string or the value based on whether the number is a multiple of 3, 5, both or neither. Determine the string to return based on the following rules for an integer i :

- If i is a multiple of both 3 and 5, print *FizzBuzz*.
- If i is a multiple of 3 (but not 5), print *Fizz*.
- If i is a multiple of 5 (but not 3), print *Buzz*.
- For all others, print the value of i .

Function Description

Complete the function `fizzBuzz` in the editor below. The function print the appropriate response for each value $i \in \{1, 2, \dots, n\}$ in ascending order, each on a separate line. There is no return value expected.

`fizzBuzz` has the following parameter(s):

n : upper limit of values to test (inclusive)

Constraints

- $0 < n < 2 \times 10^5$

[► Input Format for Custom Testing](#)[► Sample Case 0](#)

Question - 2

Braces

SCORE: 50 points

Easy

Data Structures

Algorithms

Stacks

Core Skills

Problem Solving

You are designing a compiler for a C++ program and need to check that braces in any given file are balanced.

Braces in a string are considered to be balanced if the following criteria are met:

- All braces must be closed. Braces come in pairs of the form `()`, `{}` and `[]`. The left brace *opens* the pair, and the right one *closes* it.
- In any set of nested braces, the braces between any pair must be closed.

For example, `[{}]` is a valid grouping of braces but `[]{}]` is not.

Function Description

Complete the function *braces* in the editor below. The function must return an array of strings where the string at each index *i* denotes whether or not the braces were balanced in a *values_i*. The array should consist of strings "YES" or "NO" aligned with their indexes in *values*.

braces has the following parameter(s):

values[*values₀*,...*values_{n-1}*]: an array of strings to analyze

Constraints

- $1 \leq n \leq 15$
- $1 \leq \text{length of } values_i \leq 100$
- It is guaranteed that each *values_i* consists of (,), {, }, [, and] only.

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

```
2
{}[]()
{[]}]
```

Sample Output

```
YES
NO
```

Explanation

values₀: {}[]() meets the criteria for a balanced string, so index 0 in our return array should contain the string YES.

values₁: {[]}] contains unmatched braces between a matched pair in the substrings [], {}, and [], so index 1 in our return array should contain the string NO.

Question - 3

SCORE: 75 points

Movie Titles

REST API

GET

Back-End Development

Medium

JSON

Role Based Skills

To solve this challenge, you are required to write an *HTTP GET* method to retrieve information from a movie database.

Function Description

Given a string *substr*, *getMovieTitles* must perform the following tasks:

1. Query <https://jsonmock.hackerrank.com/api/movies/search/?Title=substr> (replace *substr*).
2. Initialize the *titles* array to store **total** string elements. Store the **Title** of each movie meeting the search criterion in the *titles* array.
3. Sort *titles* in ascending order and return it as your answer.

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

- **page**: The current page.
- **per_page**: The maximum number of results per page.
- **total**: The total number of movies in the search result.
- **total_pages**: The total number of pages which must be queried to get all the results.

- `data`: An array of JSON objects containing movie information where the `Title` field denotes the title of the movie.

In order to get all results, you may have to make multiple page requests. To request a page by number, your query should read `https://jsonmock.hackerrank.com/api/movies/search/?Title=substr&page=pageNumber`, replacing `substr` and `pageNumber`.

► Input Format For Custom Testing

▼ Sample Case 0

Sample Input 0

```
spiderman
```

Sample Output 0

```
Amazing Spiderman Syndrome
Fighting, Flying and Driving: The Stunts of
Spiderman 3
Hollywood's Master Storytellers: Spiderman Live
Italian Spiderman
Spiderman
Spiderman
Spiderman 5
Spiderman and Grandma
Spiderman in Cannes
Superman, Spiderman or Batman
The Amazing Spiderman T4 Premiere Special
The Death of Spiderman
They Call Me Spiderman
```

Explanation 0

For this example, we want *all* the movie titles containing the substring `spiderman`. The response for the query `https://jsonmock.hackerrank.com/api/movies/search/?Title=spiderman&page=1` is:

```
{
  "page": "1",
  "per_page": 10,
  "total": 13,
  "total_pages": 2,
  "data": [
    {
      "Poster": "https://images-na.ssl-images-
amazon.com/images/M/MV5BYjFhN2RjZTctMzA2Ni00NzE2L
WJmYjMtNDYyTl10TkyMmY3XkEyXkFqcGdeQXVyNTA0OTU0OT
Q@._V1_SX300.jpg",
      "Title": "Italian Spiderman",
      "Type": "movie",
      "Year": 2007,
      "imdbID": "tt2705436"
    },
    {
      "Poster": "https://images-na.ssl-images-
amazon.com/images/M/MV5BMjQ4MzcwNDU3N15BM15BanBnX
kFtZTgwOTE1MzMxNzE@._V1_SX300.jpg",
      "Title": "Superman, Spiderman or Batman",
      "Type": "movie",
      "Year": 2011,
      "imdbID": "tt2084949"
    },
    {
      "Poster": "N/A",
```

```

    "Title": "Spiderman",
    "Type": "movie",
    "Year": 1990,
    "imdbID": "tt0100669"
  },
  {
    "Poster": "N/A",
    "Title": "Spiderman",
    "Type": "movie",
    "Year": 2010,
    "imdbID": "tt1785572"
  },
  {
    "Poster": "N/A",
    "Title": "Fighting, Flying and Driving: The
Stunts of Spiderman 3",
    "Type": "movie",
    "Year": 2007,
    "imdbID": "tt1132238"
  },
  {
    "Poster": "http://ia.media-
imdb.com/images/M/MV5BMjE3Mzg0MjAxMl5BMl5BanBnXkF
tZTcwNjIyODg5Mg@@._V1_SX300.jpg",
    "Title": "Spiderman and Grandma",
    "Type": "movie",
    "Year": 2009,
    "imdbID": "tt1433184"
  },
  {
    "Poster": "N/A",
    "Title": "The Amazing Spiderman T4 Premiere
Special",
    "Type": "movie",
    "Year": 2012,
    "imdbID": "tt2233044"
  },
  {
    "Poster": "N/A",
    "Title": "Amazing Spiderman Syndrome",
    "Type": "movie",
    "Year": 2012,
    "imdbID": "tt2586634"
  },
  {
    "Poster": "N/A",
    "Title": "Hollywood's Master Storytellers:
Spiderman Live",
    "Type": "movie",
    "Year": 2006,
    "imdbID": "tt2158533"
  },
  {
    "Poster": "N/A",
    "Title": "Spiderman 5",
    "Type": "movie",
    "Year": 2008,
    "imdbID": "tt3696826"
  }
]
}

```

The response for the query

[https://jsonmock.hackerrank.com/api/movies/search/?](https://jsonmock.hackerrank.com/api/movies/search/?Title=spiderman&page=2)

Title=spiderman&page=2 is:

```

{
  "page": "2",
  "per_page": 10,
  "total": 13,
  "total_pages": 2,
  "data": [
    {

```

```

"Poster": "N/A",
"Title": "They Call Me Spiderman",
"Type": "movie",
"Year": 2016,
"imdbID": "tt5861236"
},
{
  "Poster": "N/A",
  "Title": "The Death of Spiderman",
  "Type": "movie",
  "Year": 2015,
  "imdbID": "tt5921428"
},
{
  "Poster": "https://images-na.ssl-images-
amazon.com/images/M/MV5BZDlmMGQwYmItNTNmOS00OTNkL
TkxNTYtNDM3ZWVlMWUyZDIzXkEyXkFqcGdeQXVyMTA5Mzk5Mw
@@._V1_SX300.jpg",
  "Title": "Spiderman in Cannes",
  "Type": "movie",
  "Year": 2016,
  "imdbID": "tt5978586"
}
]
}

```

The values of the `Title` field for each movie in each response page in the order received are:

```

Italian Spiderman
Superman, Spiderman or Batman
Spiderman
Spiderman
Fighting, Flying and Driving: The Stunts of
Spiderman 3
Spiderman and Grandma
The Amazing Spiderman T4 Premiere Special
Amazing Spiderman Syndrome
Hollywood's Master Storytellers: Spiderman Live
Spiderman 5
They Call Me Spiderman
The Death of Spiderman
Spiderman in Cannes

```

We then sort the array in ascending order, and return it as our answer.

Question - 4

SCORE: 75 points

Music

Implementation

Medium

Algorithms

Core Skills

Problem Solving

Mark likes to listen to music while travelling. His iPod™ contains N songs and he wants to listen to L (not necessarily different) songs during a trip.

So he creates a playlist such that:

- Every song is played at least once.
- A song can be played again only if at least K other songs have been played

Mark wants to know how many different playlists are possible. Can you help Mark determine this number? As the number can be very large, display number modulo $1,000,000,007$, or (10^9+7) .

For example, if $N = 3$, $K = 1$, and $L = 3$, there are 6 playlists where all of the songs are played at least once and repeats don't occur before 1 other song has been played:

$[1,2,3]$, $[1,3,2]$, $[2,1,3]$, $[2,3,1]$, $[3,1,2]$, and $[3,2,1]$.

You are given N , K and L . You have to complete the function with the following function signature:

```
int numOfPlaylists(int N, int K, int L) {  
}
```

Constraints

- N lies between 1 and 100 , inclusive.
- K lies between 0 and N , inclusive.
- L lies between N and 100 , inclusive.

Sample Input #00:

```
1  
0  
3
```

Sample Output #00:

```
1
```

Explanation #00:

$N = 1$, so there is only 1 song in the iPod™. $K = 0$ so the song can be played as often as you want. $L = 3$, and the only valid 3-song playlist is: {song_1, song_1, song_1}.

Sample Input #01:

```
1  
1  
3
```

Sample Output #01:

```
0
```

Explanation #01:

Again, there is only 1 song in the iPod™, but it cannot be played twice in a row because $K = 1$. No valid playlists can be generated that are longer than 1 which is less than the requested $L = 3$.