

 **Odd Numbers!**

Given two integers, l and r , print all the odd numbers between l and r (l and r inclusive).

1

Complete the `oddNumbers` function in the editor below. It has 2 parameters:

2

1. An integer, l , denoting the left part of the range.
2. An integer, r , denoting the right part of the range.

The function must return an array of integers denoting the odd numbers between l and r .

3

Input Format

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains an integer, l , denoting the left part of the range.

The second line contains an integer, r , denoting the right part of the range.

Constraints

- $1 \leq l \leq r \leq 10^5$

Output Format

The function must return an array of integers denoting the odd numbers between l and r . This is printed to stdout by locked stub code in the editor.

Sample Input 0

```
2
5
```

Sample Output 0

```
3
5
```

Explanation 0

The value of l is 2 and value of r is 5. The odd numbers between $[2, 5]$ are 3 and 5.

Sample Input 1

**Sample Output 1**

3
5
7
9

1

2

3

Explanation 1

The value of l is 3 and value of r is 9. The odd numbers between $[3, 9]$ are 3, 5, 7 and 9.

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour.

[Start tour](#)*Draft saved 12:51 pm*

Original code

JavaScript (Node.js)





Sample Test

⌚ 03m : 48s
to test enc

1

3

2

```

4  var __input_stdin = "";
5  var __input_stdin_array = "";
6  var __input_currentline = 0;
7
8  process.stdin.on('data', function (data) {
9      __input_stdin += data;
10 });
11
12 /*
13  * This is my implementation that works fine when
14  * test case is run through Jasmine
15  */
16 function oddNumbers(l, r) {
17     if (l>r) return ''; // edge case check
18     var limit=100000, // including limit of 10 to power of 5 (one
    hundred thousand)
19     out=[]; // adding elements to output array
20     for (var x=l; x<=r; x++) {
21         if (x<=limit && x%2) {
22             out.push(x);
23         }
24     }
25     //console.log(out);
26     return ''+out.join("\n");
27 }
28
29 var fs = require('fs');
30 var wstream = fs.createWriteStream(process.env.OUTPUT_PATH);
31 process.stdin.on('end', function () {
32     __input_stdin_array = __input_stdin.split("\n");
33     var res;
34
35     var _l =
    parseInt(__input_stdin_array[__input_currentline].trim(), 10);
36     __input_currentline += 1;
37
38     var _r =
    parseInt(__input_stdin_array[__input_currentline].trim(), 10);
39     __input_currentline += 1;
40
41     res = oddNumbers(_l, _r);
42     for(var res_i=0; res_i < res.length; res_i++) {
43         wstream.write(res[res_i]+"\n");
44     }
45

```



Sample Test

⌚ 03m : 48s
to test enc

(You can submit any number of times)

 [Download sample test cases](#)*The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

1

2

3

Status: No test cases passed. **Tip: Debug your code against custom input****Testcase 1: Wrong Answer**Input [Download](#)2
5**Your Output**3

5Expected Output [Download](#)3
5**Testcase 2: Wrong Answer**Input [Download](#)3
9**Your Output**3

5

7



Sample Test

⌚ 03m : 48s
to test enc

1

2

3

3
5
7
9**Testcase 3: Wrong Answer**Input [\[Download\]](#)96
97**Your Output**9
7Expected Output [\[Download\]](#)

97



Sample Test

🕒 03m : 48s
to test end



1

2

3