

03m:48s



☆ 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

• $1 \le l \le r \le 10^5$

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			
5			

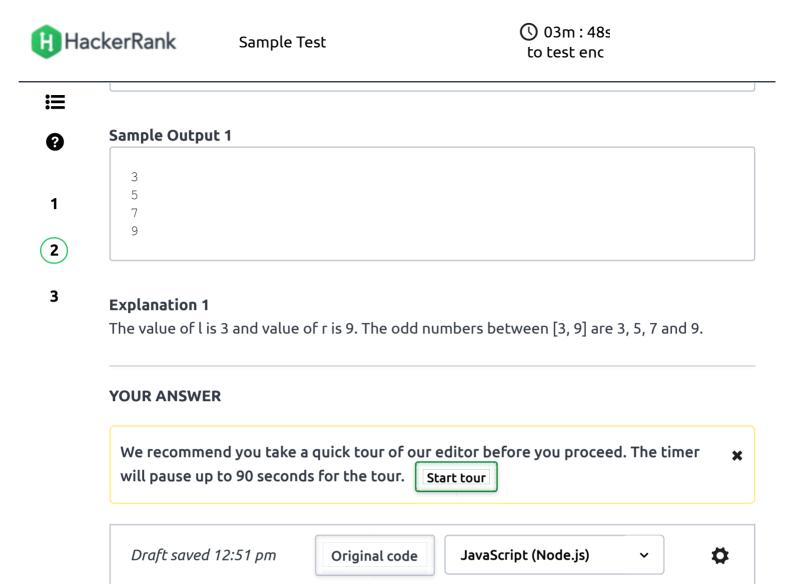
Sample Output 0

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





O3m: 48s

```
input stdin = "";
            var
三
         5
            var input stdin array = "";
            var input currentline = 0;
         6
0
         7
         8 ▼ process.stdin.on('data', function (data) {
                input stdin += data;
1
        10
            });
        11
        12 ▼ /*
2
        13
             * This is my implementation that works fine when
             * test case is run through Jasmine
        14
3
        15
        16 ▼ function oddNumbers(l, r) {
        17
                if (l>r) return ''; // edge case check
                var limit=100000, // including limit of 10 to power of 5 (one
        18
            hundred thousand)
        19
                    out=[]; // adding elements to output array
        20 ▼
                for (var x=1; 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
            var fs = require('fs');
        29
        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);
                input currentline += 1;
        36
        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++) {</pre>
        43 ▼
                    wstream.write(res[res i]+"\n");
        44
                }
        45
```



© 03m: 48s to test enc



(You can submit any number of times)



Lownload sample test cases The input/output files have Unix line endings. Do not use Notepad to edit them on windows.

1



3

Status: No test cases p	assed.
-------------------------	--------

Tip: Debug your code against custom input

Testcase 1: Wrong Answer

Input [Download]

2 5

Your Output

3

5

Expected Output [**Download**]

3 5

Testcase 2: Wrong Answer

Input [♣ Download]

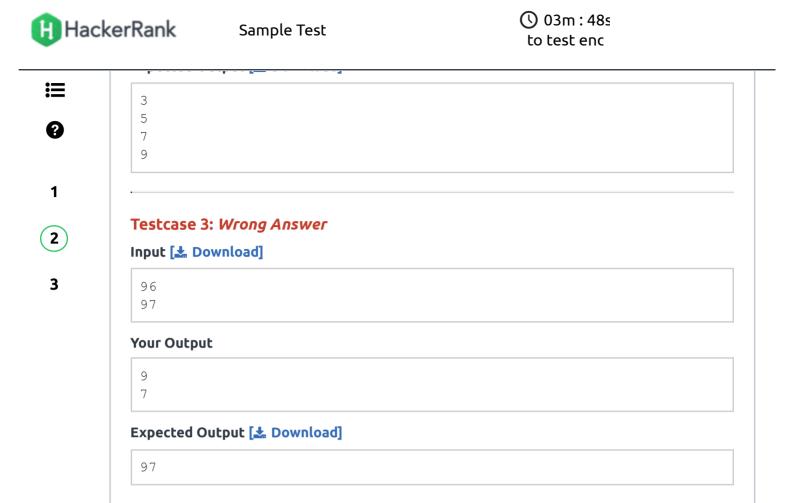
3

Your Output

3

5

7





O3m: 48s to test enc





1



3