HackerRank Sample Test

15m : 58s
to test end

C

☆ Odd Numbers!

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

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

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.

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
```

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
```

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 ✖

1
2
3

Draft saved 04:14 pm Original code Python 2

```
#!/bin/python

import sys
import os


# Complete the function below.

def  oddNumbers(l, r):
    if l%2 == 0:
        l = l + 1

    odds = []
    while l <= r:
        odds.append(l)
        l = l + 2

    return odds


f = open(os.environ['OUTPUT_PATH'], 'w')

```

HackerRank Sample Test

15m : 58s
to test end

```
_r = int(raw_input());

res = oddNumbers(_l, _r)
for res_cur in res:
    f.write( str(res_cur) + "\n" )

f.close()

```

Line: 18 Col: 13

□ Test against custom input

Run Code

Submit code & Continue

(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.*

Compiled successfully. All available test cases passed!

Tip: Debug your code against custom input

Test Case #1: ✓

Test Case #2: ✓

Test Case #3: ✓

Test Case #4: ✓

Test Case #5: ✓

Test Case #6: ✓

Test Case #7: ✓

Test Case #8: ✓

Test Case #9: ✓

Test Case #10: ✓

Test Case #11: ✓

Test Case #12: ✓

Test Case #13: ✓

```
2
5
```

Your Output

```
3
5
```

Expected Output [⤓ Download]

```
3
5
```

Testcase 2: *Success*

Input [⤓ Download]

```
3
9
```

Your Output

```
3
5
7
9
```

Expected Output [⤓ Download]

```
3
5
7
9
```

Testcase 3: *Success*

Input [⤓ Download]

```
96
97
```

Your Output

```
97
```

Expected Output [⤓ Download]

```
97
```

Testcase 4: *Success*

Testcase 5: *Success*

Your Output

```
Output hidden
```

Testcase 6: *Success*

Your Output

```
Output hidden
```

Testcase 7: *Success*

Your Output

```
Output hidden
```

Testcase 8: *Success*

Your Output

```
Output hidden
```

Testcase 9: *Success*

Your Output

```
Output hidden
```

Testcase 10: *Success*

Your Output

```
Output hidden
```

Testcase 11: *Success*

Your Output

```
Output hidden
```

Testcase 12: *Success*

Testcase 13: *Success*

Your Output

```
Output hidden
```