Twins

locked

by shef_2318

Problem | Submissions | Leaderboard | Discussions | Editorial

Submitted 11 hours ago • Score: 19.68 Status: Accepted

✔ Test Case #0	✔ 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	✔ Test Case #14
✔ Test Case #15	✔ Test Case #16	✔ Test Case #17
✔ Test Case #18	✔ Test Case #19	✔ Test Case #20
✔ Test Case #21	✔ Test Case #22	✔ Test Case #23
✔ Test Case #24	✔ Test Case #25	✔ Test Case #26
✔ Test Case #27		

Submitted Code

Language: Python 2 Open in editor

```
# Enter your code here. Read input from STDIN. Print output to STDOUT

import math
#from itertools import *
import itertools


def _try_composite(a, d, n, s):
    if pow(a, d, n) == 1:
```

```
    if pow(a, d, n) == 1:
        return False
    for i in range(s):
        if pow(a, 2**i * d, n) == n-1:
            return False
    return True # n  is definitely composite

def is_prime(n, _precision_for_huge_n=16):
    if n in _known_primes or n in (0, 1):
        return True
    if any((n % p) == 0 for p in _known_primes):
        return False
    d, s = n - 1, 0
    while not d % 2:
        d, s = d >> 1, s + 1
    # Returns exact according to http://primes.utm.edu/prove/prove2_3.html
    if n < 1373653:
        return not any(_try_composite(a, d, n, s) for a in (2, 3))
    if n < 25326001:
        return not any(_try_composite(a, d, n, s) for a in (2, 3, 5))
    if n < 118670087467:
        if n == 3215031751:
            return False
        return not any(_try_composite(a, d, n, s) for a in (2, 3, 5, 7))
    if n < 2152302898747:
        return not any(_try_composite(a, d, n, s) for a in (2, 3, 5, 7, 11))
    if n < 3474749660383:
        return not any(_try_composite(a, d, n, s) for a in (2, 3, 5, 7, 11, 13))
    if n < 341550071728321:
        return not any(_try_composite(a, d, n, s) for a in (2, 3, 5, 7, 11, 13, 17))
    # otherwise
    return not any(_try_composite(a, d, n, s)
                   for a in _known_primes[:_precision_for_huge_n])

_known_primes = [2, 3]
_known_primes += [x for x in range(5, 1000, 2) if is_prime(x)]

#print is_prime(31, _precision_for_huge_n=16)

a,b = raw_input().strip().split(' ')
n = int(a)
m = int(b)

if n==1: n+=1

ans =0
for i in range(n,m+1):
    if i + 2 <= m:
        if is_prime(i) and is_prime(i+2):
            ans+=1
            #print i , " " , (i+2)
print ans

```