Question **1**

Correct

Marked out of 10.00

You are given Q queries and in each query, there are two numbers L and R. You have to calculate the number of strong primes present in the range L and R inclusive.

Note

A strong prime is a prime number that is greater than the arithmetic mean of the nearest prime above and below. Algebraically, a prime P said to be strong if

$2P_n > P_{n-1} + P_{n+1}$,

where n is their index in the ordered set of prime numbers, where Pi denotes the i^{th} prime.

Input format

· The first line of the Input contains an integer Q denoting the number of Queries.

· Then Q lines follow each containing two numbers L and R.

Output format

· For each query, print the number of Strong primes present in the range L to R inclusively. The answer to each test case should come in a new line.

Constraint:

· $1<=Q<=10^5$

· $1<=L<=R<=10^6$

Sample Input:

3

10 20

20 30

30 50

Sample Output:

2

1

2

Explanation:

For the range 10 to 20, there are 2 strong primes (11, 17).

For the range 20 to 30, there is 1 strong prime (29).

For the range 30 to 50, there are 2 strong primes (37, 41).

For example:

Input	Result
3 10 20 20 30 30 50	2 1 2

Answer: (penalty regime: 0 %)

```
q=int(input())
max=10**6+10
prime=[1]*max
prime[0]=prime[1]=0
for i in range(2,int(max**0.5)+1):
    if prime[i]:
        for j in range(i*i,max,i):
            prime[j]=0
primes=[]
```

```
for i in range(max):
    if prime[i]:
        primes.append(i)
strong=[0]*max
for i in range(1,len(primes)-1):
    p=primes[i]
    if 2*p>primes[i-1]+primes[i+1]:
        strong[p]=1
prefix=[0]*max
for i in range(1,max):
    prefix[i]=prefix[i-1]+strong[i]
for k in range(q):
    l,r=map(int,input().split())
    print(prefix[r]-prefix[l-1])
```

	Input	Expected	Got	
✔	3 10 20 20 30 30 50	2 1 2	2 1 2	✔

Passed all tests! ✔