Question #79850

Consider the following function *h*.

```
def h(n):
        f = 0
        for i in range(1,n+1):
                if n%i == 0:
                        f = f + 1
        return(f%2 == 1)
```

The function *h(n)* given above returns True for a positive number n whenever:

n is a multiple of 2

n is a composite number

n is a prime number

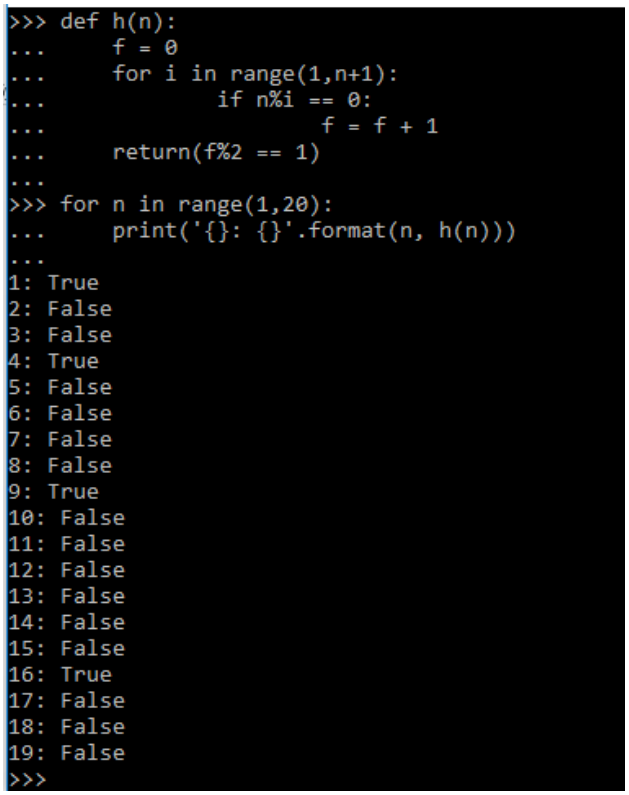
n is a perfect square

Answer:

The given function *h(n)* counts the number of all possible composition of two integer numbers, which product gives *n*, and returns *True* if the result is odd.

This algorithm is used to examine whether an integer number *n* is a perfect square.

The screenshot bellow shows the output for positive integers below 20. As we can see, *h(n)* returns *True* only for numbers 1, 4, 9 and 16, which are perfect squares.



```
>>> def h(n):
...     f = 0
...     for i in range(1,n+1):
...             if n%i == 0:
...                     f = f + 1
...     return(f%2 == 1)
...
>>> for n in range(1,20):
...     print('{}: {}'.format(n, h(n)))
...
1: True
2: False
3: False
4: True
5: False
6: False
7: False
8: False
9: True
10: False
11: False
12: False
13: False
14: False
15: False
16: True
17: False
18: False
19: False
>>>
```