- **EASY**
- **Max Score: 30 Points**

Strong numbers

A number N is strong if for every prime factor x that it has, X square also divides it. You are given a number a N, return `true` if it is strong, else return `false`.

Note Complete the given function. The input and output would be handled by the driver code.

Input Format

The first line contains a single integer N.

Output Format

Print the answer in a new line. `1` for true and `0` for false.

Example 1

Input

7

Output

0

Explanation

7 has a prime 7 as a prime factor. 7 square is 49. 49 does not divide 7.

Example 2

Input

72

Output

1

Explanation

The prime factors of 72 are 2 and 3. 2 square is 4, and 3 squared is 9. 4 and 9 both divide 72.

Constraints

1 <= N <= 10000

My code

```java
// n java
import java.util.*;
import java.lang.*;
import java.io.*;

class Main {
    static boolean checkprim(long n)
        {
```

```java
            long i,m=0;

m=n/2;
if(n==0||n==1)
{
 return false;
}
else{
 for(i=2;i<=m;i++)
 {
  if(n%i==0)
     {
      return false;
     }
 }
   return true;
}//end of else
      }
  public static boolean strongNumbers(long n)
  {
    //your code here
          long t=n;
          //boolean t=true;
          long i=2L;
          while(n>=i)
             {
                    if(n%i==0)
                    {
                           if(checkprim(i))
                           {
```

```java
                                    long p=(i*i);
                                    if(t%p!=0)
                                            return false;
                                    n=n/i;
                                i=1;
                            }
                        }
                    i++;
                }
            return true;
    }


    public static void main (String[] args) throws IOException {
            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        long n = Long.parseLong(br.readLine().trim());
        if(strongNumbers(n))
        System.out.println(1);
        else
        System.out.println(0);
        }
}
```