**Odd Divisors**

[maths](http://www.practice.geeksforgeeks.org/tag-page.php?tag=maths&isCmp=0)

Given a natural number n, print **count of numbers from 1 to n that have odd number of divisors**.  For example,  4 has odd number of divisors (1, 2, 4), but 8 doesn't (1, 2, 4, 8)

**Input:**  
The first line of input contains an integer T denoting the number of test cases. The first line of each test case is n.  
  
**Output:**  
Count of numbers from 1 to n with odd divisors.  
  
**Constraints:**  
1 <= T <= 30  
1 <= N <= 100000  
  
**Example:  
Input:**  
3  
1  
4  
5

**Output:**  
1  
2  
2

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=520>

#include <iostream>

#include <stdio.h>

#include <math.h>

#define ll long long int

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n;

scanf("%d", &n);

int answer = 0;

for(int i =1; i<=n; i++){

int cont\_divisores=0;

for(int j = 1; j<=i; j++) {

if(i%j==0){

cont\_divisores++;

}

}

if(cont\_divisores %2 != 0) {

answer++;

}

//printf("%d", i);

}

printf("%d\n", answer);

}

system("pause");

return 0;

}