**Euler Totient Sum and Divisors**

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

The**Euler Totient**Function for a positive integer N is defined as the number of positive integers less than or equal to N and relatively prime to N.  
  
For example, an algorithm to find **Euler Totient** Function value of N will be:  
  
*int phi(unsigned int N)  
{  
unsigned int result = 1;  
for (int i=2; i < N; i++)  
if (gcd(i, N) == 1)  
result++;  
return result;  
}*

The task is to find the **sum** of the **Euler Totient** Values of all the divisors of the given number.  
  
**Input:**  
The first line of input contains a single integer **T** denoting the number of test cases. Then **T** test cases follow. Each test case consists of a single line containing a positive integer **N**.  
  
**Output:**  
Corresponding to each test case, in a new line, print the sum of the Euler Totient Function values of all the divisors of the given number.

**Constraints:**  
1 ≤ T ≤ 10000  
1 ≤ N ≤ 1000000  
  
**Example:**  
**Input**  
2  
1  
2  
  
**Output**  
1  
2

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=880>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication157;

import java.util.Scanner;

public class JavaApplication157 {

public static void main(String[] args) {

// TODO code application logic here

Scanner sc = new Scanner(System.in);

int t = Integer.parseInt(sc.nextLine());

while(t-->0){

int n = Integer.parseInt(sc.nextLine());

System.out.println(n);

}

}

}