**Sherlock and his Enemies**

Show Topic Tags   

There are N different streets near the 221B Baker Street. All streets are numbered from 1 to N. Detective Lestrade consults Sherlock Homes to count the no. of gangsters living in the area, with the help of Sherlock’s underground network. Sherlock finds out that the number of gangster living in a particular street is equal to the number of 1’s in the binary representation of the street number. As Mr. Lestrade is weak in arithmetic, help him to find out the number of gangsters living in the area.

**Input:**  
The first line of input contains a integer T, denoting the number of test cases. The only line of each test case contains the integer N, described above.

**Output:**  
For each test case output a single integer corresponding to the number of gangsters living in the area, in new line.

**Constraints:**  
1<= T <=1000  
1<= N <= 106

**Example:  
Input:**  
2  
3  
4

**Output:**  
4  
5  
  
**Explanation:  
In test case 1**  
Total number of streets is 3, so Binary representation of 1,2,3 is

1-> 001, no of 1’s is 1

2-> 010, no of 1’s is 1

3-> 011, no of 1’s is 2

So the Total number of gangsters around 221B Baker Street is 4.

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

<http://practice.geeksforgeeks.org/problems/sherlock-and-his-enemies/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication245;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication245 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

int[] cont = new int[1000000];

for (int i = 1; i < cont.length; i++)

{

int total = 0;

int copia = i;

while (copia > 0)

{

if (copia % 2 == 1)

{

total++;

}

copia /= 2;

}

cont[i] += cont[i-1]+ total;

}

while(t-- > 0) {

int n = Integer.parseInt(br.readLine().trim());

System.out.println(cont[n]);

}

}

}