**Set Bits**

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

[Adobe](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Adobe&isCmp=1)[Brocade](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Brocade&isCmp=1)[Cisco](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Cisco&isCmp=1)[Juniper Networks](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Juniper%20Networks&isCmp=1)

Given a positive integer **N**, print count of set bits in it. For example, if the given number is 6, output should be 2 as there are two set bits in it.

**Input:**

The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. The next T lines will contain an integer **N**.  
  
**Output:**  
Corresponding to each test case, in a new line, print count of set bits in it.  
  
**Constraints:**

1 ≤ T ≤ 100

1 ≤ N ≤ 1000000

**Example:**

**Input:**

2  
6  
11

**Output:**  
2  
3

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1380>

#include <iostream>

#include <stdio.h>

#include <algorithm>

#define ll long long int

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n;

scanf("%d", &n);

int ans =0;

while(n > 0) {

if(n %2 ==1) {

ans++;

}

n/=2;

}

cout << ans << endl;

}

return 0;

}