Problem G
The Number of 2's
Time limit: 1 second

How many factors of 2 can you extract from factorial of n? For example, you know $10! = 1 \times 2 \times \cdots \times 10 = 3628800$. By successively extracting factor 2 from 10!, you get the following sequence

$$3628800, 1814400, 907200, 453600, 226800, 113400, 56700, 28350, 14175\ .$$

The sequence finally stops at 14175 because it is not a multiple of 2. Since there are 9 numbers in this sequence, you can get totally 8 factor 2's from 10!. We use $\epsilon_2(n!)$ to denote this number. In this example, it tells you that $\epsilon_2(10!) = 8$.

The first approach to find $\epsilon_2(n)$ is to write a computer program that computes $n!$ and then extracts 2 successively from it. However, this approach is not practical because $n!$ is easy to be very large. Thus, a more elegant way is to derive a *formula* that can compute this. Fortunately, in many textbooks about number theory, you can find such an amazing formula

$$\epsilon_2(n!) = \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{8} \right\rfloor + \cdots = \sum_{k \geq 1} \left\lfloor \frac{n}{2^k} \right\rfloor\ ,$$

where $\lfloor x \rfloor$ returns the largest integer that does not exceed x. In our example, $\epsilon_2(10!) = \lfloor 5 \rfloor + \lfloor 2.5 \rfloor + \lfloor 1.25 \rfloor = 5 + 2 + 1 = 8$ by using this formula.

The third way, possibly the most efficient one, is based on the function $v_2(n)$ that computes the number of 1's in the binary representation of n. For example, 10 is 1010_2 in binary, and there are two 1's in its binary representation. Thus, $v_2(10) = 2$. A surprising fact is $\epsilon_2(n!) = n - v_2(v)$. Based on it, you can easily calculate $\epsilon_2(10!) = 10 - v_2(10) = 10 - 2 = 8$. The computation of $v_2(n)$ can be done without using any division. Several acceleration techniques can be applied to speed up the computation of $v_2(n)$. Therefore, the third approach is possibly the most efficient way to computer $\epsilon_2(n!)$. On input n, your task is to output the value of $\epsilon_2(n!)$. Notice that either the second or third method can lead to a correct implementation that passes this task.

Input File Format

The first line gives you the total number of test cases, which is a positive integer that does not exceed 1000. Each test case occupies a single line, which contains only one positive integer n, where $1 \leq n < 2^{31}$. Test cases are listed consecutively, so there is no empty line between two adjacent cases.

Output Format

For each test case, output the value of $\epsilon_2(n!)$ in one line.

Sample Input

```
4
1
5
10
16
```

Output for the Sample Input

```
0
3
8
15
```