

Right Shift

All the numbers in a computer is represented as 64-bit 2's complement form.

You have to write a program to perform the following task :-

- Read the number (given in decimal form).
- Shift all the bits towards right (the first bit is removed), i.e the second bit from right is shifted to first position, third to second and so on.
- Add a zero to the last position.
- Write the result back in decimal form

For example 10 is represented as:

```
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1010
```

After step 2 the result is:

```
_000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101
```

After step 3 the result is:

```
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0101
```

Finally the output is: 5

Input

The first line contains **T** representing the number of test cases ($T \leq 500000$). Then T lines follows each containing a input number.

Output

Print T lines, each containing the result of each test case.

Constraints

All input and output numbers will fit in signed 64-bit integer. Large I/O. A fast code written in fast language is likely to pass.

Example

Input:

```
5
1
2
3
4
5
```

Output:

0
1
1
2
2