Mr Wireless (Digital Logic Design)

Mr. Wire Less is not that good at implementing circuit in a breadboard. In his Digital Logic Design course, he has to implement several boolean functions using the breadboard. In a breadboard, inputs are given through the switches and outputs are taken through the LEDs. Each input switch can be either in ground state or in high state. So, if he wishes to implement a boolean function, **$f(x_1, x_2, .., x_n)$** that takes **n** boolean values as input and returns **m** boolean values as output, he will use **n** switches and **m**LEDs.

Mr. Wire Less can quickly assemble the necessary ICs and wires, but the key problem is testing. While testing he has to check with all possible input combination, to make sure whether the output of LED corresponds to the expected output or not. This is taking too long for him, as most of the switches are jammed and difficult to toggle.

Mr. Wire Less is asking for help to minimize his time of testing. So, your task is to minimize the total number of switch-toggle throughout the testing.

For example, if Mr. Wire Less has to test a function **f(x0, x1)** of two variables, he may choose this switching-sequence for testing **00, 11, 10, 01**. In this case, the total number of switch-toggle will be **2+1+2 = 5**. But if he tests in this sequence **00, 10, 11, 01** total number of toggle will be **1+1+1 = 3**.

Given **n**, you have to output the minimum number of toggle needed for complete testing. Though it seems impractical, he wants you to solve the problem for a very large value of **n**. But, then the toggle value can be quite big. So, he is completely okay with the toggle value modulo **8589934592 (2^{33})**.

□□

Input

The first line of the input contains a positive integer **$T(T \le 10^5)$**, denoting the number of test-case.

Each of the following **T** lines contains a single non-negative integer **$n(n \le 10^{20})$**.

Output

For every test-case, output a single containing test-case number and the minimum number of switch-toggle modulo **8589934592 (2^{33})**.

Sample

Input

```
2
1
2
```

Output

```
Case 1: 1
Case 2: 3
```