

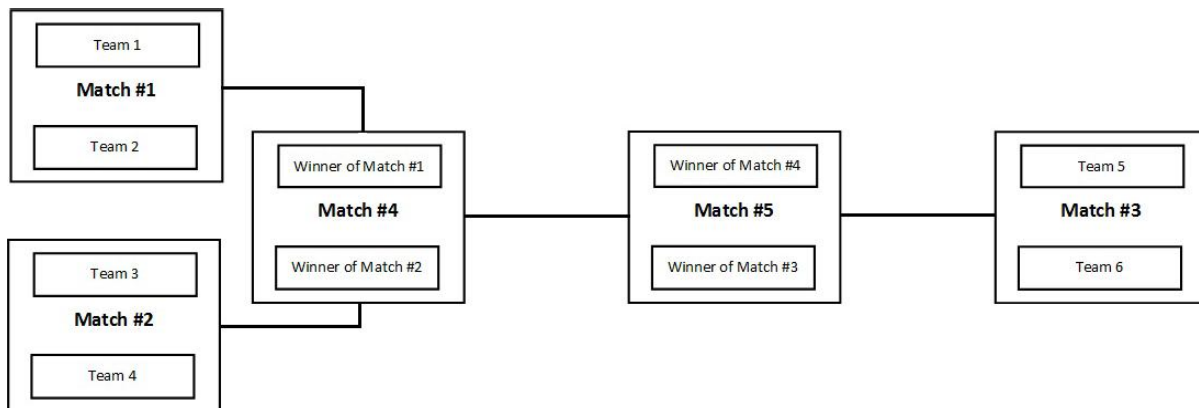
**Problem B:**  
**The League**

Source file: league.{c | cpp | java}

Input file: league.in

In a knockout stage of a football league, the loser of a match is out of the league and the winner advances to play with another winner of a different match. This method is adopted in the FIFA World Cup (which Germany recently won over Argentina) and many other leagues around the world. There are no draws in this stage, the game continues until a winner is decided (whether in the extra time or by shots from the penalty mark).

For the purposes of this problem, a league can contain any number of teams. Your task is to organize the matches between the teams to decide a winner while minimizing the number of matches. For instance, if a league has 6 teams, the figure below shows one way to organize the matches resulting in 5 matches to decide a winner of the league.



Given the number of teams in a league, your task is to determine the minimum number of matches required to decide a winner.

### Input

The first line of the input file contains an integer  $N$  ( $0 < N < 10,000$ ) indicating the number of test cases. Each test case consists of one line containing an integer  $T$  ( $0 < T \leq 10^9$ ) representing the number of teams in the league.

### Output

The output for each test case is in this form:

**k. M**

where  $k$  represents the test case number (starting at 1) followed by a single space, and  $M$  is the minimum number of matches required.

### Sample Input

```
2
6
8
```

### Output for Sample Input

```
1. 5
2. 7
```