# Problem B. Power Swapper



| | |
|---|---|
| Small input<br>4 points | Solve B-small |
| Large input<br>12 points | Solve B-large |

## Problem

In a parallel universe, people are crazy about using numbers that are powers of two, and they have defined an exciting sorting strategy for permutations of the numbers from 1 to $2^{\mathbf{N}}$. They have defined a swapping operation in the following way:

- A range of numbers to be swapped is valid if and only if it is a range of adjacent numbers of size $2^k$, and its starting position (position of the first element in the range) is a multiple of $2^k$ (where positions are 0-indexed).
- A valid swap operation of *size-k* is defined by swapping two distinct, valid ranges of numbers, each of size $2^k$.

To sort the given permutation, you are allowed to use at most one swap operation of each size k, for k in [0, **N**). Also, note that swapping a range with itself is not allowed.

For example, given the permutation [3, 6, 1, 2, 7, 8, 5, 4] (a permutation of the numbers from 1 to $2^3$), the permutation can be sorted as follows:

- [3, 6, 1, 2, 7, 8, 5, 4]: make a *size-2* swap of the ranges [3, 6, 1, 2] and [7, 8, 5, 4].
- [7, 8, 5, 4, 3, 6, 1, 2]: make a *size-0* swap of [5] and [3].
- [7, 8, 3, 4, 5, 6, 1, 2]: make a *size-1* swap of [7, 8] and [1, 2].
- [1, 2, 3, 4, 5, 6, 7, 8]: done.

The previous steps used every swap size (0, 1, and 2) at most once. Also, notice that all the swaps were valid because both ranges for each size k started at indices that were multiples of $2^k$.

Count how many ways there are to sort the given permutation by using the rules above. A way is an ordered sequence of swaps, and two ways are the same only if the sequences are identical.

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. The first line of each test case contains a single integer **N**. The following line contains $\mathbf{2^N}$ space-separated integers: a permutation of the numbers 1, 2, ..., $2^{\mathbf{N}}$.

## Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the number of ways to sort the given permutation using the rules above.

## Limits

$1 \le \mathbf{T} \le 200$.

## Small dataset

$1 \le \mathbf{N} \le 4$.

## Large dataset

$1 \le \mathbf{N} \le 12$.

## Sample

| Input | Output |
|---|---|
| 4<br>1<br>2 1<br>2<br>1 4 3 2<br>3<br>7 8 5 6 1 2 4 3 | Case #1: 1<br>Case #2: 3<br>Case #3: 6<br>Case #4: 0 |

```
2
4 3 2 1
```