

Phil is learning a new algorithm which wasn't taught in his algorithms classes. However, he is not sure whether he implemented it the right way, so he would really appreciate if you could implement it so that he can compare the outputs.

The algorithm starts with a binary string A and an empty string S . The algorithm consists of multiple steps. In each step, A and S are modified as follows:

- If the number of bits in A is odd, then the middle bit of A is added to the end of S and removed from A .
- If the number of bits in A is even, then both middle bits of A are compared. The bigger one (anyone in case of a tie) is added to the end of S and removed from A .
- If after some step the string A gets empty, the algorithm terminates. The algorithm's return is the decimal representation of the number represented by S .

A bit a is bigger than a bit b if a is 1 and b is 0.

Input

The first line contains T ($T \leq 500$) — the number of test cases, after this line T test cases follows. Each test case consists of one line containing a binary string A ($1 \leq |A| \leq 10^5$), representing the algorithm's input.

Output

For each case print a line containing 'Case # X : Y ', where X is the case number, starting at 1, and Y is the algorithm's return for the given input, modulo 1000000007 ($10^9 + 7$).

Sample Input

```
3
00000
0101101
100
```

Sample Output

```
Case #1: 0
Case #2: 106
Case #3: 2
```