

Typewriter

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

lh8k is a person who enjoys scavenging. One day, he discovered a typewriter. This typewriter has a button and two paper slots. After some experimentation, he figured out how this typewriter works.

1. At the beginning, you need to insert two pieces of paper into the two slots, with one piece of paper in the upper slot containing text as a template, and a blank piece of paper in the lower slot. The typewriter will read the content from the paper in the upper slot and write it onto the paper in the lower slot. For convenience, we denote the content of the paper in the upper slot as the string T , and the content in the lower slot as S , initially $S = \varepsilon$ (empty).
2. The typewriter has a pointer in both the upper and lower slots. We denote the positions of the pointers in the upper and lower slots as p and q , respectively, initially both pointing to the start of the two pieces of paper, i.e., $p = q = 1$.
3. Each time the button is pressed, the typewriter reads the text from the position of the pointer in the upper slot and writes that text into the position pointed to by the pointer in the lower slot (i.e., $S[q] := T[p]$). After printing, both pointers in the slots will “move” *. The pointer in the lower slot always moves forward ($q := q + 1$); the pointer in the upper slot also moves forward initially, but if the current pointer position is at the end of T , it will move backward, continuing to move until it reaches the beginning of T , then it will move forward again, repeating this cycle.

For example, if lh8k inserts a string $T = \text{abcd}$ into the upper slot and presses the button 20 times, he will obtain a printed paper in the lower slot with the string $S = \text{abcdcbabcdcbabcdcbab}$. Furthermore, if $|T| = 1$, the printed S will consist of only one character, i.e., S consists of $T[1]$ repeated several times.

lh8k now wants to print a string S using this typewriter, and he wants the length of T to be as short as possible. He wants to know what the minimum possible length of T can be. Note that the paper cannot be changed from the start to the end of the printing process, nor can the positions of the paper or pointers be changed arbitrarily. Printing S must start from the beginning of T in a forward direction.

However, for this problem, lh8k feels that just finding one answer is not enough, so he wants to find the shortest length of T for each prefix S' of the string S .

Due to the overall length of the strings being too long, for each string, you only need to output the value of $\bigoplus_{i=1}^{|S|} i \times ans_i$, where ans_i represents the shortest length of T corresponding to the prefix of length i , and \bigoplus denotes the bitwise XOR operation.

Input

The first line contains an integer t ($1 \leq t \leq 10^3$), indicating the number of test cases.

For each test case, there is one line containing a string S composed only of lowercase English letters ($1 \leq |S| \leq 10^6$), representing the string that lh8k wants to print using the typewriter.

The data guarantees that $\sum |S| \leq 10^6$.

Output

For each test case, output a single integer on a new line, representing the value of $\bigoplus_{i=1}^{|S|} i \times ans_i$, where the meaning of ans_i is as described in the problem statement.

*Actually, it is the paper that moves.

Example

standard input	standard output
5	1
a	3
aa	1
ababa	92
abcdcbabcdcbabcdcbab	51
popipopi	