Educational Codeforces Round 70 (Rated for Div. 2)

A. You Are Given Two Binary Strings...

2 seconds, 256 megabytes

You are given two binary strings x and y, which are binary representations of some two integers (let's denote these integers as $f(x)$ and $f(y)$). You can choose any integer $k \geq 0$, calculate the expression $s_k = f(x) + f(y) \cdot 2^k$ and write the binary representation of s_k in **reverse order** (let's denote it as rev_k). For example, let $x = 1010$ and $y = 11$; you've chosen $k = 1$ and, since $2^1 = 10_2$, so $s_k = 1010_2 + 11_2 \cdot 10_2 = 10000_2$ and $rev_k = 00001$.

For given x and y, you need to choose such k that rev_k is **lexicographically minimal** (read notes if you don't know what does "lexicographically" means).

It's guaranteed that, with given constraints, k exists and is finite.

Input

The first line contains a single integer T ($1 \leq T \leq 100$) — the number of queries.

Next $2T$ lines contain a description of queries: two lines per query. The first line contains one binary string x, consisting of no more than 10^5 characters. Each character is either `0` or `1`.

The second line contains one binary string y, consisting of no more than 10^5 characters. Each character is either `0` or `1`.

It's guaranteed, that $1 \leq f(y) \leq f(x)$ (where $f(x)$ is the integer represented by x, and $f(y)$ is the integer represented by y), both representations don't have any leading zeroes, the total length of x over all queries doesn't exceed 10^5, and the total length of y over all queries doesn't exceed 10^5.

Output

Print T integers (one per query). For each query print such k that rev_k is lexicographically minimal.

input
4 1010 11 10001 110 1 1 1010101010101 11110000
output
1 3 0 0

The first query was described in the legend.

In the second query, it's optimal to choose $k = 3$. The $2^3 = 1000_2$ so $s_3 = 10001_2 + 110_2 \cdot 1000_2 = 10001 + 110000 = 1000001$ and $rev_3 = 1000001$. For example, if $k = 0$, then $s_0 = 10111$ and $rev_0 = 11101$, but $rev_3 = 1000001$ is lexicographically smaller than $rev_0 = 11101$.

In the third query $s_0 = 10$ and $rev_0 = 01$. For example, $s_2 = 101$ and $rev_2 = 101$. And 01 is lexicographically smaller than 101.

The quote from Wikipedia: "To determine which of two strings of characters comes when arranging in *lexicographical order*, their first letters are compared. If they differ, then the string whose first letter comes earlier in the alphabet comes before the other string. If the first letters are the same, then the second letters are compared, and so on. If a position is reached where one string has no more letters to compare while the other does, then the first (shorter) string is deemed to come first in alphabetical order."