Problem L. Levenshtein Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

The **Levenshtein Distance** between two strings is the smallest number of simple one-letter operations needed to change one string to the other. The operations are:

- Adding a letter anywhere in the string.
- Removing a letter from anywhere in the string.
- Changing any letter in the string to any other letter.

You will be given a number k and two strings S and T. Your task is to find the number of non-empty substrings of T whose Levenshtein Distance between S is exactly i for every possible non-negative integer i ($0 \leq i \leq k$). Two substrings are considered different if and only if they occur in different places.

Input

The first line contains a single integer k ($0 \leq k \leq 30$), denoting the parameter k.

The second line contains a string S ($1 \leq |S| \leq 10^5$), denoting the pattern string.

The third line contains a string T ($1 \leq |T| \leq 10^5$), denoting the text string.

It is guaranteed that the input strings only consist of lowercase English letters ('`a`' to '`z`'), uppercase English letters ('`A`' to '`Z`'), and digits ('`0`' to '`9`').

Output

Output $k+1$ lines, the i-th ($1 \leq i \leq k+1$) of which contains an integer denoting the number of substrings of T whose Levenshtein Distance between S is exactly $i-1$.

Example

standard input	standard output
4 aaa aabbaab	0 5 15 7 1