Problem 4 - Data Structures and Anticheating (100 pts)

Problem Description

Story

After prior attempts in developing games, Nextgen Technology Universal Company of Software (NTUCS) unfortunately lost too much money and decided to use its excellence in Data Structures and Algorithm technology to develop other kinds of software.

Meanwhile, cheating is unfortunately found in homework on Dense Structures of Asphalt (DSA) class in New Tempura University (NTU). It is suspected that a pair of submissions appeared to be cheating.

A case of cheating usually involves a source submission and a copier submission. Each submission can be represented by a string consisting of lower-case letters, i.e., $\Sigma = \{a, b, \ldots, x, y, z\}$, except in subtask 2, where $\Sigma = \{a\}$. Thus, in the following, we will use the term "submission" and "string" interchangeably.

First, we define *hit count* $\mathfrak{H}(P, Q)$ of an input pattern P and a reference string Q as the number of occurrences of P in Q. This is the same as finding the number of valid shifts in the string matching problem, given by

$$\mathfrak{H}(P, Q) = |\{s : 0 \leq s \leq |Q| - |P| \text{ and } Q[s + 1 : s + |P|] = P\}| \ .$$

Next, we define *plagiarism likelihood* $\mathfrak{L}(R)$ of a string R as the number of substrings of R such that they are both the prefix and the suffix of R. This is equivalent to finding the number of prefixes of R that are also the suffixes of R, given by

$$\mathfrak{L}(R) = |\{k : 1 \leq k \leq |R| \text{ and } R[1 : k] \sqsupset R\}| \ .$$

Note that $\mathfrak{L}(R)$ is at least 1 because the input string R itself is both the prefix and the suffix of itself.

Finally, combining these two definitions, we can define the *similarity score* $\mathfrak{S}(I, S)$ of an input string I and a source submission S as follows:

$$\mathfrak{S}(I, S) = \mathfrak{H}(I, S) \cdot \mathfrak{L}(I) \ .$$

Nextgen Technology Universal Company of Software does not have a sufficient number of software engineers so they ask for your help. In this problem, for a given pair of copier submission C and source submission S, please output the similarity scores of all prefixes of C (as input string) and source submission S as follows: $\mathfrak{S}(C[1 : 1], S), \mathfrak{S}(C[1 : 2], S), \ldots, \mathfrak{S}(C[1 : |C|], S)$.

Input

The first line has the string S representing the source submission. The second line has the string C representing the copier submission.

Output

Print $|C|$ lines. In the i-th line, $1 \leq i \leq |C|$, output the similarity score $\mathfrak{S}(C[1:i], S)$.

Constraints

- $0 < |S| \leq 5 \times 10^6$
- $0 < |C| \leq 5 \times 10^6$
- Time Limit: 3 s
- Memory Limit: 262144 KB

Subtasks

Subtask 1 (10 pts)

- $|S| \leq 100$

Subtask 2 (10 pts)

- The character set Σ to form S and C only has character 'a', i.e., $\Sigma = \{a\}$ and $|\Sigma| = 1$.

Subtask 3 (20 pts)

- $|S| \leq 5000$

Subtask 4 (60 pts)

- No other constraints.

Sample Testcases

Sample Input 1	Sample Output 1
aaaaa aaa	5 8 9

Explanation for Sample 1

- Prefix `a`: The **hit count** is 5 as `a` appeared in `aaaaa` 5 times. The **plagiarism likelihood** is 1 since `a` is both a prefix and a suffix. The **similarity score** is thus 5.
- Prefix `aa`: The **hit count** is 4 as `aa` appeared in `aaaaa` 4 times. The **plagiarism likelihood** is 2 since `a` and `aa` are both prefixes and suffixes. The **similarity score** is thus 8.
- Prefix `aaa`: The **hit count** is 3 as `aaa` appeared in `aaaaa` 3 times. The **plagiarism likelihood** is 3 since `a`, `aa` and `aaa` are all prefixes and suffixes. The **similarity score** is thus 9.
- This input satisfies the constraint of subtask 2.

Sample Input 2	**Sample Output 2**
`ntucsiedsa`	`1`
`csie`	`1`
	`1`
	`1`

Explanation for Sample 2

- Prefix `c`: The **hit count** is 1 as `c` appeared in `ntucsiedsa` 1 times. The **plagiarism likelihood** is 1 since only `c` is both a prefix and a suffix. The **similarity score** is thus 1.
- Prefix `cs`: The **hit count** is 1 as `cs` appeared in `ntucsiedsa` 1 times. The **plagiarism likelihood** is 1 since only `cs` is both a prefix and a suffix. The **similarity score** is thus 1.
- Prefix `csi`: The **hit count** is 1 as `csi` appeared in `ntucsiedsa` 1 times. The **plagiarism likelihood** is 1 since only `csi` is both a prefix and a suffix. The **similarity score** is thus 1.
- Prefix `csie`: The **hit count** is 1 as `csie` appeared in `ntucsiedsa` 1 times. The **plagiarism likelihood** is 1 since only `csie` is both a prefix and a suffix. The **similarity score** is thus 1.

Sample Input 3	Sample Output 3
`baabaab`	`3`
`baab`	`2`
	`2`
	`4`

Explanation for Sample 3

- Prefix `b`: The **hit count** is 3 as `b` appeared in `baabaab` 3 times. The **plagiarism likelihood** is 1 since only `b` is both a prefix and a suffix. The **similarity score** is thus 3.
- Prefix `ba`: The **hit count** is 2 as `ba` appeared in `baabaab` 2 times. The **plagiarism likelihood** is 1 since only `ba` is both a prefix and a suffix. The **similarity score** is thus 2.
- Prefix `baa`: The **hit count** is 2 as `baa` appeared in `baabaab` 2 times. The **plagiarism likelihood** is 1 since only `baa` is both a prefix and a suffix. The **similarity score** is thus 2.
- Prefix `baab`: The **hit count** is 2 as `baab` appeared in `baabaab` 2 times. The **plagiarism likelihood** is 2 since `b` and `baab` are both prefixes and suffixes. The **similarity score** is thus 4.

Hints

- Prefix = Suffix? Sounds familiar, doesn't it?