Problem H
Longest Substring

Time Limit: 5.0 Seconds

For a string S of length $n \geq 1$ and a positive integer k ($1 \leq k \leq n$), a non-empty substring of S is called a *k-substring* if the substring appears *exactly* k times. Such k occurrences are not necessarily disjoint, i.e., are possibly overlapping. For example, if S = "ababa", the k-substrings of S for every $k = 1, \dots, 5$ are as follows:

- There are four 1-substrings in S, "abab", "ababa", "bab", and "baba" because these substrings appear exactly once in S. Note that "aba" is not a 1-substring because it appears twice.
- There are four 2-substrings in S, "ab", "aba", "b", and "ba". The substring "ab" appears exactly twice without overlapping. Two occurrences of the substring "aba" are overlapped at a common character "a", but it does not appear three times or more.
- There is only one 3-substring in S, "a".
- Neither 4-substrings nor 5-substrings exist in S.

For a k-substring T of S, let $d(T)$ be the maximum number of the disjoint occurrences of T in S. For example, a 2-substring T = "ab" can be selected twice without overlapping, that is, the maximum number of the disjoint occurrences is two, so $d(T) = 2$. For a 2-substring T = "aba", it cannot be selected twice without overlapping, so $d(T) = 1$. For a 3-substring T = "a", it can be selected three times without overlapping, which is the maximum, so $d(T) = 3$.

Let $f(k)$ be the length of the longest one among all k-substring T with the largest $d(T)$ for $1 \leq k \leq n$. For example, $f(k)$ for S = "ababa" and $k = 1, \dots, 5$ is as follows:

- For $k = 1$, all 1-substrings T can be selected only once without overlapping, so $d(T) = 1$. Thus, the longest one among all 1-substrings with $d(T) = 1$ is "ababa", so $f(1) = 5$.
- For $k = 2$, $d(T) = 1$ for T = "aba", but $d(T) = 2$ for the other 2-substrings T = "ab", "b", "ba". Among 2-substrings with $d(T) = 2$, "ab" and "ba" are the longest ones, so $f(2) = 2$.
- For $k = 3$, $f(3) = 1$ because there is only one 3-substring "a".
- For $k = 4, 5$, there are no k-substrings, so $f(4) = 0$ and $f(5) = 0$.

Given a string S of length n, write a program to output n values of $f(k)$ from $k = 1$ to $k = n$. For the above example, the output should be 5 2 1 0 0.

Input

Your program is to read from standard input. The input starts with a line containing the string S consisting of n ($1 \leq n \leq 50{,}000$) lowercase alphabets.

Output

Your program is to write to standard output. Print exactly one line. The line should contain exactly n non-negative integers, separated by a space, that represent $f(k)$ from $k = 1$ to $k = n$ in order, that is, $f(1)\ f(2)\ \dots\ f(n)$. Note that $f(k)$ should be zero if there is no k-substring for some k.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
`ababa`	`5 2 1 0 0`

Sample Input 2	Output for the Sample Input 2
`aaaaaaaa`	`8 7 6 5 4 3 2 1`