

Problem I. Suffix Sort

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 1024 mebibytes

Grammy has a string S of length n that consists of lowercase English letters.

For a string P , reading it left to right, write down the letters that never occurred before as $t_1, t_2, t_3, \dots, t_k$. For example, if $P = \text{"sesame"}$, we write down 's', 'e', 'a', 'm'. The *minimal representation* $R(P)$ can be obtained by replacing every occurrence of t_1 in P by the first character of the character set ("a"), replacing every occurrence of t_2 in P by the second character of the character set ("b"), and so on.

For example, when the character set is lowercase English letters, the minimal representation of "sesame" is "abacdb", the minimal representation of "edcca" is "abccd", and minimal representations $R(\text{"xy"})$ and $R(\text{"zt"})$ are both "ab".

Your task is to sort all suffixes of S by their minimal representation. Formally, denote suffix $S_i S_{i+1} \dots S_{n-1} S_n$ as $S[i:]$. For two suffixes $S[i:]$ and $S[j:]$, if $R(S[i:])$ is less than $R(S[j:])$ in lexicographical order, then $S[i:]$ has to occur before $S[j:]$ in the desired order.

Please output the result as an array of indices sa : the i -th element of $sa[i]$ must be the position of the first character in the i -th smallest suffix of S in the desired order. Formally, the array must satisfy

$$R(S[sa[1]:]) < R(S[sa[2]:]) < \dots < R(S[sa[n]:]).$$

Input

The first line contains one integers n ($1 \leq n \leq 200\,000$).

The next line contains a string S of length n . It is guaranteed that S only consists of lowercase English alphabets.

Output

Output n integers representing the answer.

Example

standard input	standard output
6 aadead	6 1 5 4 3 2