Pseudo-Isomorphic Substrings

Two strings A and B, consisting of small English alphabet letters are called pseudo-isomorphic if

- Their lengths are equal
- For every pair (i,j), where 1 <= i < j <= |A|, B[i] = B[j], iff A[i] = A[j]
- For every pair (i,j), where 1 <= i < j <= |A|, B[i] != B[j] iff A[i] != A[j]

Naturally, we use 1-indexation in these definitions and |A| denotes the length of the string **A**.

You are given a string S, consisting of no more than 10^5 lowercase alphabetical characters. For every prefix of **S** denoted by S', you are expected to find the size of the largest possible set of strings , such that all elements of the set are substrings of S' and no two strings inside the set are pseudo-isomorphic to each other.

if S = abcde
then, 1st prefix of S is 'a'
then, 2nd prefix of S is 'ab'
then, 3rd prefix of S is 'abc'
then, 4th prefix of S is 'abcd' and so on..

Input Format

The first and only line of input will consist of a single string S. The length of S will not exceed 10^5.

Constraints

- $1 \leq |S| \leq 10^5$
- S contains only lower-case english alphabets ('a' - 'z').

Output Format

Output N lines. On the i^{th} line, output the size of the largest possible set for the first i alphabetical characters of S such that no two strings in the set are pseudo-isomorphic to each other.

Sample Input

```
abbabab
```

Sample Output

```
1
2
4
6
9
12
15
```

Explanation

The first character is 'a', the set is {a} hence 1.
The first 2 characters are 'ab', the set is {a, b, ab} but 'a' is pseudo-isomorphic to 'b'. So, we can remove either 'a' or 'b' from the set. We get {a,ab} or {b,ab}, hence 2.
Similarly, the first 3 characters are 'abb', the set is {a, ab, abb, b, bb} and as 'a' is pseudo-isomorphic to

'b', we have to remove either 'a' or 'b' from the set. We get {a,ab, abb, bb}, hence 4. and so on...