ACM Algorithm Practice **Spring 2015**

Week 1 Problems

April 3, 2015 *Timothy Johnson*

Problem 1: Strings of Power

(Taken from http://codeforces.com/problemset/problem/318/B)
Time limit: 2 seconds

Volodya likes listening to heavy metal and (occasionally) reading. No wonder Volodya is especially interested in texts concerning his favourite music style.

Volodya calls a string powerful if it starts with "heavy" and ends with "metal". Finding all powerful substrings (by substring Volodya means a subsequence of consecutive characters in a string) in a given text makes our hero especially joyful. Recently he felt an enormous fit of energy while reading a certain text. So Volodya decided to count all powerful substrings in this text and brag about it all day long. Help him in this difficult task. Two substrings are considered different if they appear at the different positions in the text.

For simplicity, let us assume that Volodya's text can be represented as a single string.

Input: Input contains a single non-empty string consisting of the lowercase Latin alphabet letters. The length of this string will not be greater than 10^6 characters.

Output: Print exactly one number the number of powerful substrings of the given string.

Examples:

- Input: heavymetalisheavymetal
  Output: 3
- Input: heavymetalismetal
  Output: 2
- Input: heavymetalheavymetalheavymetalheavymetal
  Output: 10

In the first sample the string "heavymetalisheavymetal" contains the powerful substring "heavymetal" twice, also the whole string "heavymetalisheavymetal" is certainly powerful.

In the second sample the string "heavymetalismetal" contains two powerful substrings: "heavymetal" and "heavymetalismetal".

Problem 2: Worms

(Taken from http://codeforces.com/problemset/problem/474/B)
Time limit: 1 second

It is lunch time for Mole. His friend, Marmot, prepared him a nice game for lunch.

Marmot brought Mole n ordered piles of worms such that the i-th pile contains a_i worms. He labeled all these worms with consecutive integers: worms in the first pile are labeled from 1 to a_1, worms in the second pile are labeled from $a_1 + 1$ to $a_1 + a_2$, and so on.

Mole can't eat all the worms (Marmot brought a lot) and, as we all know, Mole is blind, so Marmot tells him the label of the best juicy worms. Marmot will only give Mole a worm if Mole says correctly in which pile this worm is located.

Poor Mole asks for your help. For all juicy worms said by Marmot, tell Mole the correct pile.

Input: The first line contains a single integer n ($1 \leq n \leq 10^5$), the number of piles.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^3$, $a_1 + a_2 + \ldots + a_n \leq 10^6$), where a_i is the number of worms in the i-th pile.

The third line contains a single integer m ($1 \leq m \leq 10^5$), the number of juicy worms said by Marmot.

The fourth line contains m integers $q_1, q_2 \ldots, q_m$ ($1 \leq q_i \leq a_1 + a_2 + \ldots + a_n$), the labels of the juicy worms.

Output: Print m lines to standard output. The i-th line should contain an integer, representing the number of the pile that contains the i-th juicy worm.

Example:
Input:
5
2 7 3 4 9
3
1 25 11

Output:
1
5
3

For the sample input:

- The worms from [1,2] are in the first pile.
- The worms from [3,9] are in the second pile.

- The worms from [10,12] are in the third pile.
- The worms from [13,16] are in the fourth pile.
- The worms from [17,25] are in the fifth pile.

For a harder version of this problem, increase the maximum number of worms so that $1 \leq a_i \leq 10^6$, $1 \leq a_1 + a_2 + \ldots + a_n \leq 10^{11}$.

Problem 3: No to Palindromes!

(Taken from http://codeforces.com/problemset/problem/464/A)
Time limit: 1 second

Paul hates palindromes. He assumes that string s is tolerable if each its character is one of the first p letters of the English alphabet and s doesn't contain any palindrome contiguous substring of length 2 or more.

Paul has found a tolerable string s of length n. Help him find the lexicographically next tolerable string of the same length or else state that such a string does not exist.

Input:
The first line contains two space-separated integers, n and p ($1 \leq n \leq 1000$, $1 \leq p \leq 26$). The second line contains string s, consisting of n lower-case English letters. It is guaranteed that the string is tolerable (by the above definition).

Output:
If the lexicographically next tolerable string of the same length exists, print it. Otherwise print "NO" (without the quotes).

Examples:

- Input:
  3 3
  cba

  Output:
  NO

- Input:
  3 4 cba

  Output:
  cbd

- Input:
  4 4
  abcd

  Output:
  abda