

2340 - ACM Ordered Partitions

Description

In general, an ordered partition of a set T of type (m_1, m_2, \dots, m_k) where m_i belongs to \mathbb{Z}^+ , is a sequence of disjoint sets (B_1, B_2, \dots, B_k) such that $|B_i| = m_i$, $i = 1, 2, \dots, k$, and $\bigcup_{i=1}^k B_i = T$. Empty sets are allowed in ordered partitions. The distinct letters in the word *MISSISSIPPI* (base) are I, M, S, and P. There are eleven letter positions in the word *MISSISSIPPI* which we can explicitly label as follows:

p { margin-bottom: 0.08in; }

1	2	3	4	5	6	7	8	9	1	1
									0	1
<i>M</i>	<i>I</i>	<i>S</i>	<i>S</i>	<i>I</i>	<i>S</i>	<i>S</i>	<i>I</i>	<i>P</i>	<i>P</i>	<i>I</i>

We can describe this placement of letters by a rule such as: **I {2, 5, 8, 11}, M {1}, P {9, 10}, and S {3, 4, 6, 7}**. If we remember the ordering (alphabetic in this case), I, M, P, S, then we can specify this arrangement by the ordered partition **({2, 5, 8, 11}, {1}, {9, 10}, {3, 4, 6, 7})** of the set **{1, 2, ..., 11}**. We say that this ordered partition is of type **(4, 1, 2, 4)**, referring to the sizes of the sets, in order, that make up the ordered partition. The set of all rearrangements of the letters in the word *MISSISSIPPI* corresponds to the set of all ordered partitions **(B₁, B₂, B₃, B₄)** of **{1, 2, ..., 11}** of type **(4, 1, 2, 4)**.

For example the set of all rearrangements of the letters in the word *MISSISSIPPI* corresponds to the set of all ordered partitions **(B₁, B₂, B₃, B₄)** of **{1, 2, ..., 11}** of type **(4, 1, 2, 4)**. An ordered partition **({1, 5, 7, 10}, {2}, {9, 11}, {3, 4, 6, 8})** corresponds to the placement **I {1, 5, 7, 10}, M {2}, P {9, 11}, and S {3, 4, 6, 8}** and leads to the "word":

p { margin-bottom: 0.08in; }

1	2	3	4	5	6	7	8	9	1	1
									0	1
<i>I</i>	<i>M</i>	<i>S</i>	<i>S</i>	<i>I</i>	<i>S</i>	<i>I</i>	<i>S</i>	<i>P</i>	<i>I</i>	<i>P</i>

The program will receive as input a word and an ordered list of partitions, and produce as output for each ordered partition given the corresponding word.

Input specification

The first line contains the word base (with no more than **10** uppercase letters). In the following lines are a series of partitions (at most **10**) whose subsets are ordered in different lines, and ordered partitions are separated by a blank line. They are preceded by one integer, telling you the number of partitions of the word base. Your program should be able to process a number of cases as described.

Output specification

For each case, the output consists of a list of words corresponding to the ordered sequences and there will be one word per line. You must also print a blank line, before each case.

Sample input

EXERCISES

```
1
3
4 6 8
5
2
1 7
9
```

CONTIENE

```
1
4
2 7
6
3 8
5
1
```

Sample output

SRCEIESEX

TENCOIEN

Hint(s)

Source	Claudia Guadalupe Gómez Santillán
Added by	ymondelo20
Addition date	2013-04-09
Time limit (ms)	5000
Test limit (ms)	2500
Memory limit (kb)	256000
Output limit (mb)	64
Size limit (bytes)	30000
Enabled languages	Bash C C# C++ Java Pascal Perl PHP Python Ruby Text