## 敌兵布阵

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)**

**Problem Description**

C国的死对头A国这段时间正在进行军事演习，所以C国间谍头子Derek和他手下Tidy又开始忙乎了。A国在海岸线沿直线布置了N个工兵营地,Derek和Tidy的任务就是要监视这些工兵营地的活动情况。由于采取了某种先进的监测手段，所以每个工兵营地的人数C国都掌握的一清二楚,每个工兵营地的人数都有可能发生变动，可能增加或减少若干人手,但这些都逃不过C国的监视。  
 中央情报局要研究敌人究竟演习什么战术,所以Tidy要随时向Derek汇报某一段连续的工兵营地一共有多少人,例如Derek问:“Tidy,马上汇报第3个营地到第10个营地共有多少人!”Tidy就要马上开始计算这一段的总人数并汇报。但敌兵营地的人数经常变动，而Derek每次询问的段都不一样，所以Tidy不得不每次都一个一个营地的去数，很快就精疲力尽了，Derek对Tidy的计算速度越来越不满:"你个死肥仔，算得这么慢，我炒你鱿鱼!”Tidy想：“你自己来算算看，这可真是一项累人的工作!我恨不得你炒我鱿鱼呢!”无奈之下，Tidy只好打电话向计算机专家Windbreaker求救,Windbreaker说：“死肥仔，叫你平时做多点acm题和看多点算法书，现在尝到苦果了吧!”Tidy说："我知错了。。。"但Windbreaker已经挂掉电话了。Tidy很苦恼，这么算他真的会崩溃的，聪明的读者，你能写个程序帮他完成这项工作吗？不过如果你的程序效率不够高的话，Tidy还是会受到Derek的责骂的.

**Input**

第一行一个整数T，表示有T组数据。  
每组数据第一行一个正整数N（N<=50000）,表示敌人有N个工兵营地，接下来有N个正整数,第i个正整数ai代表第i个工兵营地里开始时有ai个人（1<=ai<=50）。  
接下来每行有一条命令，命令有4种形式：  
(1) Add i j,i和j为正整数,表示第i个营地增加j个人（j不超过30）  
(2)Sub i j ,i和j为正整数,表示第i个营地减少j个人（j不超过30）;  
(3)Query i j ,i和j为正整数,i<=j，表示询问第i到第j个营地的总人数;  
(4)End 表示结束，这条命令在每组数据最后出现;  
每组数据最多有40000条命令

**Output**

对第i组数据,首先输出“Case i:”和回车,  
对于每个Query询问，输出一个整数并回车,表示询问的段中的总人数,这个数保持在int以内。

**Sample Input**

1

10

1 2 3 4 5 6 7 8 9 10

Query 1 3

Add 3 6

Query 2 7

Sub 10 2

Add 6 3

Query 3 10

End

**Sample Output**

Case 1:

6

33

59

## Queue

time limit per test

2 seconds

memory limit per test

256 megabytes

There are *n* walruses standing in a queue in an airport. They are numbered starting from the queue's tail: the 1-st walrus stands at the end of the queue and the *n*-th walrus stands at the beginning of the queue. The *i*-th walrus has the age equal to *ai*.

The *i*-th walrus becomes displeased if there's a younger walrus standing in front of him, that is, if exists such *j* (*i* < *j*), that *ai* > *aj*. The displeasure of the *i*-th walrus is equal to the number of walruses between him and the furthest walrus ahead of him, which is younger than the *i*-th one. That is, the further that young walrus stands from him, the stronger the displeasure is.

The airport manager asked you to count for each of *n* walruses in the queue his displeasure.

**Input**

The first line contains an integer *n* (2 ≤ *n* ≤ 105) — the number of walruses in the queue. The second line contains integers *ai* (1 ≤ *ai* ≤ 109).

Note that some walruses can have the same age but for the displeasure to emerge the walrus that is closer to the head of the queue needs to be **strictly younger** than the other one.

**Output**

Print *n* numbers: if the *i*-th walrus is pleased with everything, print "-1" (without the quotes). Otherwise, print the *i*-th walrus's displeasure: the number of other walruses that stand between him and the furthest from him younger walrus.

**Examples**

**Input**

**Copy**

6  
10 8 5 3 50 45

**Output**

**Copy**

2 1 0 -1 0 -1

**Input**

**Copy**

7  
10 4 6 3 2 8 15

**Output**

**Copy**

4 2 1 0 -1 -1 -1

**Input**

**Copy**

5  
10 3 1 10 11

**Output**

**Copy**

1 0 -1 -1 -1

题意：题意：给一个序列，对于第i个数字a[i]，在右边找到一个比它小（严格小）的数,并且最靠右的位置k，输出k-i-1，如果一个都找不到，输出-1。对于序列的每个元素都要输出。

## String Reconstruction

time limit per test

2 seconds

memory limit per test

256 megabytes

Ivan had string *s* consisting of small English letters. However, his friend Julia decided to make fun of him and hid the string *s*. Ivan preferred making a new string to finding the old one.

Ivan knows some information about the string *s*. Namely, he remembers, that string *ti* occurs in string *s* at least *ki* times or more, he also remembers exactly *ki* positions where the string *ti* occurs in string *s*: these positions are *xi*, 1, *xi*, 2, ..., *xi*,*ki*. He remembers *n* such strings *ti*.

You are to reconstruct **lexicographically minimal** string *s* such that it fits all the information Ivan remembers. Strings *ti* and string *s* consist of small English letters only.

**Input**

The first line contains single integer *n* (1 ≤ *n* ≤ 105) — the number of strings Ivan remembers.

The next *n* lines contain information about the strings. The *i*-th of these lines contains non-empty string *ti*, then positive integer *ki*, which equal to the number of times the string *ti* occurs in string *s*, and then *ki* distinct positive integers *xi*, 1, *xi*, 2, ..., *xi*,*ki* in increasing order — positions, in which occurrences of the string *ti* in the string *s* start. It is guaranteed that the sum of lengths of strings *ti* doesn't exceed 106, 1 ≤ *xi*,*j* ≤ 106, 1 ≤ *ki* ≤ 106, and the sum of all *ki* doesn't exceed 106. The strings *ti* can coincide.

It is guaranteed that the input data is not self-contradictory, and thus at least one answer **always** exists.

**Output**

Print lexicographically minimal string that fits all the information Ivan remembers.

**Examples**

**Input**

**Copy**

3  
a 4 1 3 5 7  
ab 2 1 5  
ca 1 4

**Output**

**Copy**

abacaba

**Input**

**Copy**

1  
a 1 3

**Output**

**Copy**

aaa

**Input**

**Copy**

3  
ab 1 1  
aba 1 3  
ab 2 3 5

**Output**

**Copy**

ababab