## 亲戚

若某个家族人员过于庞大，要判断两个是否是亲戚，确实还很不容易，现在给出某个亲戚关系图，求任意给出的两个人是否具有亲戚关系。

规定：x和y是亲戚，y和z是亲戚，那么x和z也是亲戚。如果x,y是亲戚，那么x的亲戚都是y的亲戚，y的亲戚也都是x的亲戚。

数据输入：

第一行：三个整数n,m,p，（n<=5000,m<=5000,p<=5000），分别表示有n个人，m个亲戚关系，询问p对亲戚关系。

以下m行：每行两个数Mi，Mj，1<=Mi，Mj<=N，表示Ai和Bi具有亲戚关系。

接下来p行：每行两个数Pi，Pj，询问Pi和Pj是否具有亲戚关系。

数据输出：

P行，每行一个’Yes’或’No’。表示第i个询问的答案为“具有”或“不具有”亲戚关系。

样例：

input.txt

6 5 3

1 2

1 5

3 4

5 2

1 3

1 4

2 3

5 6

output.txt

Yes

Yes

No

## Ubiquitous Religions(POJ2524)

来源：POJ 2524

There are so many different religions in the world today that it is difficult to keep track of them all. You are interested in finding out how many different religions students in your university believe in.

You know that there are n students in your university (0 < n <= 50000). It is infeasible for you to ask every student their religious beliefs. Furthermore, many students are not comfortable expressing their beliefs. One way to avoid these problems is to ask m (0 <= m <= n(n-1)/2) pairs of students and ask them whether they believe in the same religion (e.g. they may know if they both attend the same church). From this data, you may not know what each person believes in, but you can get an idea of the upper bound of how many different religions can be possibly represented on campus. You may assume that each student subscribes to at most one religion.

Input

The input consists of a number of cases. Each case starts with a line specifying the integers n and m. The next m lines each consists of two integers i and j, specifying that students i and j believe in the same religion. The students are numbered 1 to n. The end of input is specified by a line in which n = m = 0.

Output

For each test case, print on a single line the case number (starting with 1) followed by the maximum number of different religions that the students in the university believe in.

Sample Input

10 9

1 2

1 3

1 4

1 5

1 6

1 7

1 8

1 9

1 10

10 4

2 3

4 5

4 8

5 8

0 0

Sample Output

Case 1: 1

Case 2: 7

## Two Sets

来源：Codeforces 468B

time limit per test 1 second

memory limit per test 256 megabytes

Little X has *n* distinct integers: *p*1, *p*2, ..., *pn*. He wants to divide all of them into two sets *A* and *B*. The following two conditions must be satisfied:

* If number *x* belongs to set *A*, then number *a* - *x* must also belong to set *A*.
* If number *x* belongs to set *B*, then number *b* - *x* must also belong to set *B*.

Help Little X divide the numbers into two sets or determine that it's impossible.

**Input**

The first line contains three space-separated integers *n*, *a*, *b* (1 ≤ *n* ≤ 105; 1 ≤ *a*, *b* ≤ 109). The next line contains *n* space-separated distinct integers *p*1, *p*2, ..., *pn* (1 ≤ *pi* ≤ 109).

**Output**

If there is a way to divide the numbers into two sets, then print "YES" in the first line. Then print *n* integers: *b*1, *b*2, ..., *bn* (*bi* equals either 0, or 1), describing the division. If *bi* equals to 0, then *pi* belongs to set *A*, otherwise it belongs to set *B*.

If it's impossible, print "NO" (without the quotes).

**Examples**

**input**

**Copy**

4 5 9  
2 3 4 5

**output**

**Copy**

YES  
0 0 1 1

**input**

**Copy**

3 3 4  
1 2 4

**output**

**Copy**

NO

**Note**

It's OK if all the numbers are in the same set, and the other one is empty.

## Socks

来源：Codeforces 731C

time limit per test 2 seconds

memory limit per test 256 megabytes

Arseniy is already grown-up and independent. His mother decided to leave him alone for *m* days and left on a vacation. She have prepared a lot of food, left some money and washed all Arseniy's clothes.

Ten minutes before her leave she realized that it would be also useful to prepare instruction of which particular clothes to wear on each of the days she will be absent. Arseniy's family is a bit weird so all the clothes is enumerated. For example, each of Arseniy's *n* socks is assigned a unique integer from 1 to *n*. Thus, the only thing his mother had to do was to write down two integers *li* and *ri* for each of the days — the indices of socks to wear on the day *i* (obviously, *li* stands for the left foot and *ri* for the right). Each sock is painted in one of *k*colors.

When mother already left Arseniy noticed that according to instruction he would wear the socks of different colors on some days. Of course, that is a terrible mistake cause by a rush. Arseniy is a smart boy, and, by some magical coincidence, he posses *k* jars with the paint — one for each of *k* colors.

Arseniy wants to repaint some of the socks in such a way, that for each of *m* days he can follow the mother's instructions and wear the socks of the same color. As he is going to be very busy these days he will have no time to change the colors of any socks so he has to finalize the colors now.

The new computer game Bota-3 was just realised and Arseniy can't wait to play it. What is the minimum number of socks that need their color to be changed in order to make it possible to follow mother's instructions and wear the socks of the same color during each of *m* days.

**Input**

The first line of input contains three integers *n*, *m* and *k* (2 ≤ *n* ≤ 200 000, 0 ≤ *m* ≤ 200 000, 1 ≤ *k* ≤ 200 000) — the number of socks, the number of days and the number of available colors respectively.

The second line contain *n* integers *c*1, *c*2, ..., *cn* (1 ≤ *ci* ≤ *k*) — current colors of Arseniy's socks.

Each of the following *m* lines contains two integers *li* and *ri* (1 ≤ *li*, *ri* ≤ *n*, *li* ≠ *ri*) — indices of socks which Arseniy should wear during the *i*-th day.

**Output**

Print one integer — the minimum number of socks that should have their colors changed in order to be able to obey the instructions and not make people laugh from watching the socks of different colors.

**Examples**

**input**

**Copy**

3 2 3  
1 2 3  
1 2  
2 3

**output**

**Copy**

2

**input**

**Copy**

3 2 2  
1 1 2  
1 2  
2 1

**output**

**Copy**

0

**Note**

In the first sample, Arseniy can repaint the first and the third socks to the second color.

In the second sample, there is no need to change any colors.

## 团伙

1920年的芝加哥，出现了一群强盗。如果两个强盗遇上了，那么他们要么是朋友，要么是敌人。而且有一点是肯定的，就是：

我朋友的朋友是我的朋友；

我敌人的敌人也是我的朋友。

两个强盗是同一团伙的条件是当且仅当他们是朋友。现在给你一些关于强盗们的信息，问你最多有多少个强盗团伙。

输入格式：

第一行是一个整数N(2<=N<=1000)，表示强盗的个数（从1编号到N）。 第二行M(1<=M<=5000)，表示关于强盗的信息条数。 以下M行，每行可能是F p q或是E p q（1<=p q<=N），F表示p和q是朋友，E表示p和q是敌人。输入数据保证不会产生信息的矛盾。

输出格式：

只有一行，表示最大可能的团伙数。

输入样例：

6

4

E 1 4

F 3 5

F 4 6

E 1 2

输出样例：

3

## Destroying Array

来源：Codeforces 722C

time limit per test 1 second

memory limit per test 256 megabytes

You are given an array consisting of *n* non-negative integers *a*1, *a*2, ..., *an*.

You are going to destroy integers in the array one by one. Thus, you are given the permutation of integers from 1 to *n* defining the order elements of the array are destroyed.

After each element is destroyed you have to find out the segment of the array, such that it contains no destroyed elements and the sum of its elements is maximum possible. The sum of elements in the empty segment is considered to be 0.

**Input**

The first line of the input contains a single integer *n* (1 ≤ *n* ≤ 100 000) — the length of the array.

The second line contains *n* integers *a*1, *a*2, ..., *an* (0 ≤ *ai* ≤ 109).

The third line contains a permutation of integers from 1 to *n* — the order used to destroy elements.

**Output**

Print *n* lines. The *i*-th line should contain a single integer — the maximum possible sum of elements on the segment containing no destroyed elements, after first *i* operations are performed.

**Examples**

**input**

**Copy**

4  
1 3 2 5  
3 4 1 2

**output**

**Copy**

5  
4  
3  
0

**input**

**Copy**

5  
1 2 3 4 5  
4 2 3 5 1

**output**

**Copy**

6  
5  
5  
1  
0

**input**

**Copy**

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

**output**

**Copy**

18  
16  
11  
8  
8  
6  
6  
0

**Note**

Consider the first sample:

1. Third element is destroyed. Array is now 1 3  \*  5. Segment with maximum sum 5 consists of one integer 5.
2. Fourth element is destroyed. Array is now 1 3  \*   \* . Segment with maximum sum 4 consists of two integers 1 3.
3. First element is destroyed. Array is now  \*  3  \*   \* . Segment with maximum sum 3 consists of one integer 3.

Last element is destroyed. At this moment there are no valid nonempty segments left in this array, so the answer is equal to 0.

## 银河英雄传说

宇宙历七九九年，银河系的两大军事集团在巴米利恩星域爆发战争。泰山压顶集团派宇宙舰队司令莱因哈特率领十万余艘战舰出征，气吞山河集团点名将杨威利组织麾下三万艘战舰迎敌。

杨威利擅长排兵布阵，巧妙运用各种战术屡次以少胜多，难免恣生骄气。在这次决战中，他将巴米利恩星域战场划分成30000列，每列依次编号为1, 2, …, 30000。之后，他把自己的战舰也依次编号为1, 2, …, 30000，让第i号战舰处于第i列(i = 1, 2, …, 30000)，形成“一字长蛇阵”，诱敌深入。这是初始阵形。当进犯之敌到达时，杨威利会多次发布合并指令，将大部分战舰集中在某几列上，实施密集攻击。合并指令为M i j，含义为让第i号战舰所在的整个战舰队列，作为一个整体（头在前尾在后）接至第j号战舰所在的战舰队列的尾部。显然战舰队列是由处于同一列的一个或多个战舰组成的。合并指令的执行结果会使队列增大。

然而，老谋深算的莱因哈特早已在战略上取得了主动。在交战中，他可以通过庞大的情报网络随时监听杨威利的舰队调动指令。

在杨威利发布指令调动舰队的同时，莱因哈特为了及时了解当前杨威利的战舰分布情况，也会发出一些询问指令：C i j。该指令意思是，询问电脑，杨威利的第i号战舰与第j号战舰当前是否在同一列中，如果在同一列中，那么它们之间布置有多少战舰。

作为一个资深的高级程序设计员，你被要求编写程序分析杨威利的指令，以及回答莱因哈特的询问。

输入：

第一行有一个整数T（1<=T<=500,000），表示总共有T条指令。

以下有T行，每行有一条指令。指令有两种格式：

1.  M i j ：i和j是两个整数（1<=i , j<=30000），表示指令涉及的战舰编号。该指令是莱因哈特窃听到的杨威利发布的舰队调动指令，并且保证第i号战舰与第j号战舰不在同一列。

2. C i j ：i和j是两个整数（1<=i , j<=30000），表示指令涉及的战舰编号。该指令是莱因哈特发布的询问指令。

输出：

依次对输入的每一条指令进行分析和处理：

如果是杨威利发布的舰队调动指令，则表示舰队排列发生了变化，你的程序要注意到这一点，但是不要输出任何信息；

如果是莱因哈特发布的询问指令，你的程序要输出一行，仅包含一个整数，表示在同一列上，第i号战舰与第j号战舰之间布置的战舰数目。如果第i号战舰与第j号战舰当前不在同一列上，则输出-1。

样例输入

4

M 2 3

C 1 2

M 2 4

C 4 2

样例输出

-1

1

## 食物链

来源：POJ 1182

动物王国中有三类动物A,B,C，这三类动物的食物链构成了有趣的环形。A吃B， B吃C，C吃A。

现有N个动物，以1－N编号。每个动物都是A,B,C中的一种，但是我们并不知道它到底是哪一种。

有人用两种说法对这N个动物所构成的食物链关系进行描述：

第一种说法是"1 X Y"，表示X和Y是同类。

第二种说法是"2 X Y"，表示X吃Y。

此人对N个动物，用上述两种说法，一句接一句地说出K句话，这K句话有的是真的，有的是假的。当一句话满足下列三条之一时，这句话就是假话，否则就是真话。

1） 当前的话与前面的某些真的话冲突，就是假话；

2） 当前的话中X或Y比N大，就是假话；

3） 当前的话表示X吃X，就是假话。

你的任务是根据给定的N（1 <= N <= 50,000）和K句话（0 <= K <= 100,000），输出假话的总数。

Input

第一行是两个整数N和K，以一个空格分隔。

以下K行每行是三个正整数 D，X，Y，两数之间用一个空格隔开，其中D表示说法的种类。

若D=1，则表示X和Y是同类。

若D=2，则表示X吃Y。

Output

只有一个整数，表示假话的数目。

Sample Input

100 7

1 101 1

2 1 2

2 2 3

2 3 3

1 1 3

2 3 1

1 5 5

Sample Output

3