**Hdu 1166 敌兵布阵**

**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

**hdu 1754 I Hate It**

**Problem Description**

很多学校流行一种比较的习惯。老师们很喜欢询问，从某某到某某当中，分数最高的是多少。

这让很多学生很反感。

不管你喜不喜欢，现在需要你做的是，就是按照老师的要求，写一个程序，模拟老师的询问。当然，老师有时候需要更新某位同学的成绩。

**Input**

本题目包含多组测试，请处理到文件结束。

在每个测试的第一行，有两个正整数 N 和 M ( 0<N<=200000,0<M<5000 )，分别代表学生的数目和操作的数目。

学生ID编号分别从1编到N。

第二行包含N个整数，代表这N个学生的初始成绩，其中第i个数代表ID为i的学生的成绩。

接下来有M行。每一行有一个字符 C (只取'Q'或'U') ，和两个正整数A，B。

当C为'Q'的时候，表示这是一条询问操作，它询问ID从A到B(包括A,B)的学生当中，成绩最高的是多少。

当C为'U'的时候，表示这是一条更新操作，要求把ID为A的学生的成绩更改为B。

**Output**

对于每一次询问操作，在一行里面输出最高成绩。

**Sample Input**

5 6

1 2 3 4 5

Q 1 5

U 3 6

Q 3 4

Q 4 5

U 2 9

Q 1 5

**Sample Output**

5 6 5 9

**Hint**

Huge input,the C function scanf() will work better than cin

**poj 2299**

**题意：**

求逆序对

**POJ3468**

**Description**

You have N integers, A1, A2, ... , AN. You need to deal with two kinds of operations. One type of operation is to add some given number to each number in a given interval. The other is to ask for the sum of numbers in a given interval.

**Input**

The first line contains two numbers N and Q. 1 ≤ N,Q ≤ 100000.

The second line contains N numbers, the initial values of A1, A2, ... , AN. -1000000000 ≤ Ai ≤ 1000000000.

Each of the next Q lines represents an operation.

"C a b c" means adding c to each of Aa, Aa+1, ... , Ab. -10000 ≤ c ≤ 10000.

"Q a b" means querying the sum of Aa, Aa+1, ... , Ab.

**Output**

You need to answer all Q commands in order. One answer in a line.

**Sample Input**

10 5

1 2 3 4 5 6 7 8 9 10

Q 4 4

Q 1 10

Q 2 4

C 3 6 3

Q 2 4

**Sample Output**

4

55

9

15

**题意：**

给你N个数，Q个操作，操作有两种，‘Q a b ’是询问a~b这段数的和，‘C a b c’是把a~b这段数都加上c。

**poj2528 Mayor‘s posters**

**Description**

The citizens of Bytetown, AB, could not stand that the candidates in the mayoral election campaign have been placing their electoral posters at all places at their whim. The city council has finally decided to build an electoral wall for placing the posters and introduce the following rules:

Every candidate can place exactly one poster on the wall.

All posters are of the same height equal to the height of the wall; the width of a poster can be any integer number of bytes (byte is the unit of length in Bytetown).

The wall is divided into segments and the width of each segment is one byte.

Each poster must completely cover a contiguous number of wall segments.

They have built a wall 10000000 bytes long (such that there is enough place for all candidates). When the electoral campaign was restarted, the candidates were placing their posters on the wall and their posters differed widely in width. Moreover, the candidates started placing their posters on wall segments already occupied by other posters. Everyone in Bytetown was curious whose posters will be visible (entirely or in part) on the last day before elections.

Your task is to find the number of visible posters when all the posters are placed given the information about posters' size, their place and order of placement on the electoral wall.

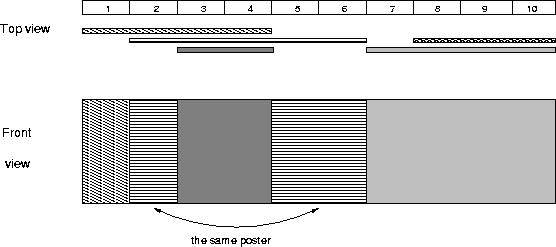
Input

The first line of input contains a number c giving the number of cases that follow. The first line of data for a single case contains number 1 <= n <= 10000. The subsequent n lines describe the posters in the order in which they were placed. The i-th line among the n lines contains two integer numbers li and ri which are the number of the wall segment occupied by the left end and the right end of the i-th poster, respectively. We know that for each 1 <= i <= n, 1 <= li <= ri <= 10000000. After the i-th poster is placed, it entirely covers all wall segments numbered li, li+1 ,... , ri.

Output

For each input data set print the number of visible posters after all the posters are placed.

The picture below illustrates the case of the sample input.



**Sample Input**

1

5

1 4

2 6

8 10

3 4

7 10

**Sample Output**

4

**题意**

给每个区间贴纸，后面贴的会覆盖前面贴的，问最后能看到的纸有几张