Problem E: Down, Right and Up!

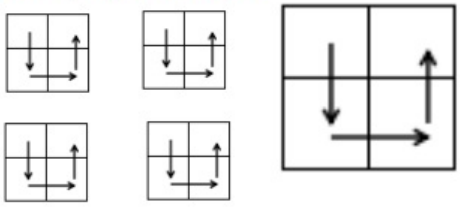
Time Limit: 4 Sec Memory Limit: 128 MB
Submit: 711 Solved: 141
[Submit (submitpage.php?cid=1052&pid=4&langmask=0)][Status (problemstatus.php?id=1249)][Web Board (bbs.php?pid=1249&cid=1052)]

Description

How to count an infinity set? For example, all the points in a 2-D space which have **positive integer** coordinates. lanran finds a way to solve that, 'Down, Right and Up!' Let's see how we count first several points by the above way.

Plane coordinates:	Indexes of those coordinates:	Go Down, Right and Up!
(1,1) (1,2)	1 4	
(2,1) (2,2)	2 3	

How do we count more?

Plane coordinates:	Indexes of those coordinates:	Go Down, Right and Up!
(1,1) (1,2) (1,3) (1,4)	1 4 13 16	
(2,1) (2,2) (2,3) (2,4)	2 3 14 15	
(3,1) (3,2) (3,3) (3,4)	5 8 9 12	
(4,1) (4,2) (4,3) (4,4)	6 7 10 11	

Hopefully, you have understood this procedure well. Similarly, we can expand the 'matrix' to infinitely large, that is, we can count all the points which have positive integer coordinates in a 2-D space.

Given a coordinate, you need to return its index and given an index, you need to return its coordinate.

Input

The first line of input is the number of test cases T (1<=T<=100,000)
For each test case, there could only be a coordinate,**(x,y)** or an index **a**.(1<=x,y<=10^9 , 1<=a<=10^{18}) You may need 'long long' in C++, or 'long' in Java to store the value of a.

Output

The coordinate or index that matches each query.

Sample Input

```
2
1
(1,2)
```

Sample Output

```
(1,1)
4
```

HINT

[Submit (submitpage.php?cid=1052&pid=4&langmask=0)][Status (problemstatus.php?id=1249)]