Little John is very interested about constructing a rooted tree with the following constraints:

1. A tree of depth D means that the tree should contain at least 1 node which is exactly D distance away from the root and there is no node of more than D distance from the root.

2. The degree of a node of the tree cannot be greater than V. Degree of a node is simply measured by the number of nodes it is directly connected to, via a single edge.

John wonders about the maximum number of nodes in a tree following the rules described above. For example, if $D = 1$ and $V = 2$, then the maximum number of nodes in the tree is 3.

Input

First line of the input contains a positive integer T ($T \leq 150$). Each of the following T lines contains two integers D ($0 \leq D \leq 2 * 10^9$) and V ($1 \leq V \leq 2 * 10^9$), respectively.

Output

For each case, print a line of the form '`Case` $< x >$`:` $< y >$', where x is the case number and y is the maximum possible number of nodes in the tree. As the value of y can be quite large, print the value *modulo* 1000000007 ($10^9 + 7$). If it is not possible to construct the tree, print '`Case` $< x >$`: -1`'.

Sample Input

```
3
0 1
1 2
1 5
```

Sample Output

```
Case 1: 1
Case 2: 3
Case 3: 6
```