Problem E

Foreign Exchange

Input: standard input
Output: standard output
Time Limit: 1 second

Your non-profit organization (**iCORE** - **i**nternational **C**onfederation of **R**evolver **E**nthusiasts) coordinates a very successful foreign student exchange program. Over the last few years, demand has sky-rocketed and now you need assistance with your task.

The program your organization runs works as follows: All candidates are asked for their original location and the location they would like to go to. The program works out only if every student has a suitable exchange partner. In other words, if a student wants to go from A to B, there must be another student who wants to go from B to A. This was an easy task when there were only about 50 candidates, however now there are up to **500000** candidates!

Input

The input file contains multiple cases. Each test case will consist of a line containing **n** - the number of candidates **(1≤n≤500000)**, followed by **n** lines representing the exchange information for each candidate. Each of these lines will contain **2** integers, separated by a single space, representing the candidate's original location and the candidate's target location respectively. Locations will be represented by nonnegative integer numbers. You may assume that no candidate will have his or her original location being the same as his or her target location as this would fall into the domestic exchange program. The input is terminated by a case where **n = 0**; this case should not be processed.

Output

For each test case, print **"YES"** on a single line if there is a way for the exchange program to work out, otherwise print **"NO"**.

Sample Input	Output for Sample Input
10 1 2 2 1 3 4 4 3 100 200 200 100 57 2 2 57 1 2 2 1 10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 0	YES NO

Problem F

Signed Digit Numbers

Input: standard input
Output: standard output
Time Limit: 2 seconds

Typically, in a base *b* number system the digit set is
{ 0, 1, ..., *b*-1 }
However, it is possible to allow a digit set with negative digits
{ -*a*, -(*a*-1), ..., -1, 0, 1, ..., (*a*-1), *a* }
with appropriately chosen *a*. In the sequel, the negative digits will be written as `'d` when `d` is a positive digit, such that a negative digit occupies two characters in a recording of a number. For example, with *b* = 10 and *a* = 6 we have the following digits
{`'6`, `'5`, `'4`, `'3`, `'2`, `'1`, `0`, `1`, `2`, `3`, `4`, `5`, `6` }
The resulting number system is called *signed-digit* system and in order to make the parameters explicit we call it *SD(b, a)* system, in the case above *SD(10, 6)*. Computing a number from its representation in an *SD* system is the same as in usual number systems, just some digits have negative values.
Using two digits in *SD(10, 6)* we can record all the numbers in the range '6'6..66, i.e. in the usual decimal notation the range is -66..66, but with the new digits we do not need a sign for the entire number. The *SD* systems are redundant in that a number can have more than one representation. For example, in *SD(10, 6)* the decimal number 4 has at least two representations: `4` and `1'6`.
The redundancy of *SD(b, a)* number system allows to design faster algorithms for addition. If the following conditions are satisfied
ceil((*b*+1)/2) <= *a* <= *b*-1
we can choose one of the available representations of each number in such a way that we can design an algorithm for addition which runs in constant time as the carry propagation can be eliminated. However, we will not be concerned with carry propagation here.
Your task is simple. Given a number *n* in usual decimal notation fitting into a 32-bit integer, a positive *b* <= 10 and *a* satisfying the conditions stated above, you are asked to convert *n* into its *SD(b, a)* representation where the negative digits are recorded as explained above. If *n* has more than one representation in *SD(b, a)*, then anyone will do.

Input

The input file contains a sequence of lines, each line contains three numbers: *n*, *b* and *a*. The last line of input has *b* = 0 and should not be processed.

Output

For each line of input produce one line of output containing a representation of *n* in *SD(b, a)*.

Sample Input	Output for Sample Input
98 10 6 -89 10 6 456789 5 3 2147483647 6 4 0 10 9 0 0 0	10'2 '111 11'111'113'1 10'13032010'131 0

Problem setter: Piotr Rudnicki, University of Alberta, Canada

Again Prime? No time.

Input: standard input
Output: standard output
Time Limit: 1 second

The problem statement is very easy. Given a number ***n*** you have to determine the largest power of ***m***, not necessarily prime, that divides ***n!***.

Input

The input file consists of several test cases. The first line in the file is the number of cases to handle. The following lines are the cases each of which contains two integers ***m (1<m<5000)*** and ***n (0<n<10000)***. The integers are separated by an space. There will be no invalid cases given and there are not more that ***500*** test cases.

Output

For each case in the input, print the case number and result in separate lines. The result is either an integer if ***m*** divides ***n!*** or a line "***Impossible to divide***" (without the quotes). Check the sample input and output format.

Sample Input

```
2
2 10
2 100
```

Sample Output

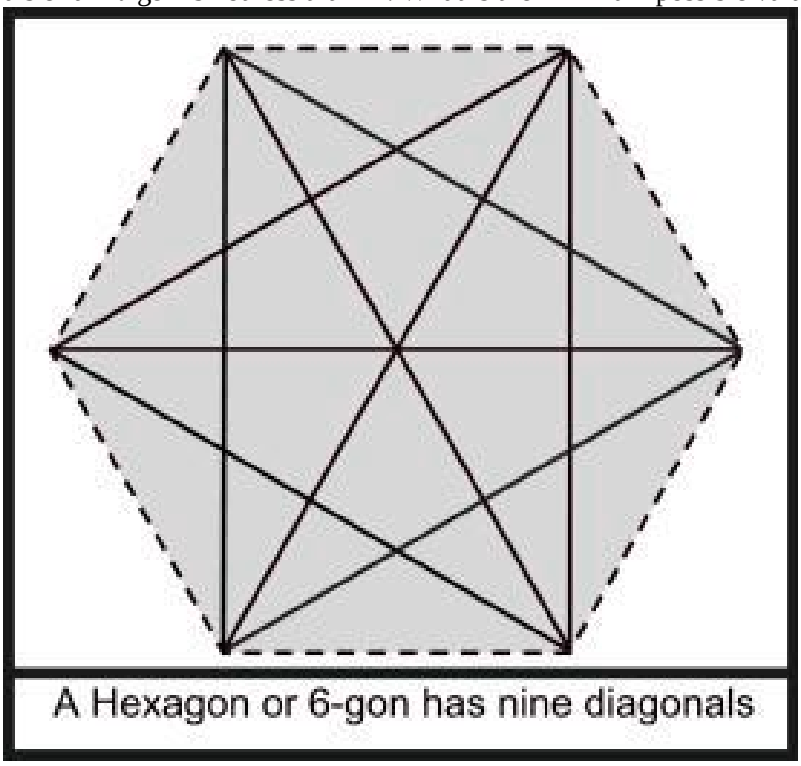
```
Case 1:
8
Case 2:
97
```

Problem setter: Anupam Bhattacharjee, CSE, BUET
Thanks to Shabuj for checking and Adrian for alternate solution.

〝～ Algorithms are the rhythms of Computer Science ～〞

Diagonal

The number of diagonals of an n-gon is not less than N. What is the minimum possible value of n?



Input

The input file contains less than 1001 lines of inputs. Each line contains a positive integer N ($N \leq 10^{15}$) that indicates the minimum possible number of diagonals. Input is terminated by a line containing a zero. This line should not be processed.

Output

For each line of input produce one line of output, which contains the output serial number, and also the minimum possible value for n (Number of sides).

Sample Input

```
10
100
1000
0
```

Sample Output

```
Case 1: 7
Case 2: 16
Case 3: 47
```

Miguel Revilla 2004-12-02