1324 – Equivalent Boolean Expressions

In this problem you are given some Boolean expressions, you can evaluate them and find whether they return true or false. A Boolean expression consists of three basic operators, and (&), or (|) or not (!). '&' and '|' are binary operators, where '!' is a unary operator. '&' returns true if and only if both of its operands are true, '|' returns true if any of its operands is true. '!' negates the operand, that means it makes true to false and vice versa. The grammar is given below:

Expression	**=>**	**Term | Expression '|' Term**
Term	**=>**	**Factor | Term '&' Factor**
Factor	**=>**	**Sub | '!' Factor**
Sub	**=>**	**Variable | '(' Expression ')'**
Variable	**=>**	**'a' | 'b' | ... | 'j'**

Table 1: Grammar in EBNF

And Boolean variables can be true or false. In this problem, you are given two Boolean expressions, your task is to find whether they are equivalent or not. Two Boolean expressions are said equivalent if they produce same results in all situations. That means if we change the value of the variables, the expressions produce same results.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains two lines. In each line there will be a non-empty string representing a Boolean expression which follows the grammar described above. The strings will not contain any white spaces and the length will be less than 101.

Output

For each case, print the case number and **'Equivalent'** if the expressions are equivalent; or **'Not Equivalent'** if they are not. Be careful about spelling and case sEnSitiVity.

Sample Input	Output for Sample Input
5 a|b !(!a&!b) !(!a|!b) a&b a|b|c a|(b&c) (a|!a)|(c&!c) !(b&!b) a|(b&c) (a&b)|(a&c)	Case 1: Equivalent Case 2: Equivalent Case 3: Not Equivalent Case 4: Equivalent Case 5: Not Equivalent