3701 John's Canonical Difference Bound Matrices

When John studied the timed automaton, he met the problem about how to trigger the machine. With the problem deeply studied, he found that it can be ascribed to the clock constraints of the timed automaton. The timed automation in question is described below:

The clock variables, or simply clocks, are variables whose values are integers. Of course, time passes at the same rate for all clocks, and any clock can be reset to zero. John uses C to denote the finite set of clocks, and defines the clock constraints for C as follows:

(1) All inequalities of the form $t\#c$ or $c\#t$ are clock constraints, where t is a clock, $\#$ is either $<$ or $\leq$, and c is an integer.

(2) If A_1 and A_2 are clock constraints, then $A_1 \wedge A_2$ is a clock constraint.

John notes that a clock constraint can define several regions in some multidimensional space. He wants to know such regions, so he defines the clock zones recursively as follows.

$$A = x < c \mid x \leq c \mid c < x \mid c \leq x \mid x - y < c \mid x - y \leq c \mid A_1 \wedge A_2$$

For simplicity, he let $C_0 = C \cup \{x_0\}$, where x_0 is a reference clock whose value is always 0. The clock zone A can be described by a Difference Bound Matrix D (called a *DBM*) which is a matrix (D_{ij}) of size $|C_0| \times |C_0|$. Each D_{ij} has the form $(d_{ij}, \#)$, where $d_{ij} \in Z \cup \{\$\}$, $\# \in \{<, \leq\}$. The value of D_{ij} can be evaluated in the following form:

For every inequality $x_i - x_j \# d_{ij}$ in clock zone A, let $D_{ij} = (d_{ij}, \#)$, where x_i and x_j are two clocks. If the bound of $x_i - x_j$ for x_i and x_j is unknown, let $D_{ij} = (\$, <)$.

For example, DBM of the clock zone given by $x_1 - x_2 < 2 \wedge 0 < x_2 \leq 2 \wedge 1 \leq x_1$ is shown below:

```
(0, <=)  (-1, <=)  (0, <)
($, <)   (0, <=)   (2, <)
(2, <=)  ($, <)    (0, <=)
```

The representation of a clock zone by a DBM is not unique. In this example, there are some implied constraints that are not reflected in the DBM. Since $x_1 - x_2 < 2$ and $x_2 \leq 2$, it must be the case $x_1 < 4$. Since $x_0 = 0$, the original $D_{10} = (\$, <)$ can be changed into $D_{10} = (4, <)$. Such adjusting operation is called the tighten operation.

Now John wants to do the similar adjusting operations of difference bounds for all clocks x_i and x_j until further application of this tighten operation does not change the matrix. John obtains the following new canonical difference bound matrix:

```
(0, <=)  (-1, <=)  (0, <)
(4, <)   (0, <=)   (2, <)
(2, <=)  (1, <)    (0, <=)
```

Note that some clock zone may contain contrary conditions and has not canonical difference bound matrix.

But John can not obtain a canonical difference bound matrix for a complex clock zone. He asks for your help.

Input

The first line of the input file is a single integer T ($1 \leq T \leq 20$), which is the number of test cases you must process, followed by T test cases:

Each test case consists of several lines. Four integers i, j, d and r are given on each line, representing a constraint $x_i - x_j < d$ or $x_i - x_j \leq d$ ($0 \leq i, j \leq m$, $-10000 < d < 10000$). If $r = 0$, then this line represents an inequality in the form of $x_i - x_j < d$, otherwise it represents an inequality in the form of $x_i - x_j \leq d$. The maximal index m of clocks indicates that the indexes of the clocks are $0, 1, \ldots, m$, ($1 \leq m \leq 100$). Note that you have to get the value of m by yourself.

A symbol '`#`' given on a single line indicates the end of a test case.

Output

For each test case, first output '`Case #:`' on a single line, where # is the case number starting from 1. Print a blank line after each test case.

For each test case, output the description of the canonical difference bound matrix. If it doesn't have a canonical difference bound matrix, print '`Canonical DBM does not exist.`' (without quotes); If it has a such canonical difference bound matrix, print the matrix in the format as indicated in the sample output. Every element D_{ij} of the matrix should be written in the form $(d_{ij}, \#)$, where # is either '`<`' or '`<=`'. If the bound of $x_i - x_j$ for x_i and x_j is unknown, print '`($,<)`' at the position (i, j). Two consecutive elements on each row should be separated by a single space.

Sample Input

```
1
1 2 2 0
0 2 0 0
2 0 2 1
0 1 -1 1
#
```

Sample Output

```
Case 1:
(0,<=) (-1,<=) (0,<)
(4,<) (0,<=) (2,<)
(2,<=) (1,<=) (0,<=)
```