

Fantasy Chess

Lea loves chess. After having mastered the classical game completely, she turned her attention to fantasy chess, with huge boards and special pieces. Currently she studies the strategy of using the knight-like figures, that can make a hop of a fields in one direction and b fields in an orthogonal direction. An important part is of course finding a way to reach a field with coordinates larger by (x, y) from the original field. Can you write a program for that?

Input

The first line of the input contains an integer t . t test cases follow.

Each test case is a line containing four integers a, b, x, y . Here a and b specify the move of the piece, and x and y specify the target field.

Output

For each test case, output one line containing “Case # i :” where i is its number, starting at 1. If there is a possibility to reach the target cell, output up to eight more lines, containing three integers a', b', k each. a' and b' should be a and b in some order and possibly negated. Each such line says that the piece should make k moves (a', b') . The number k must be small enough that $a \cdot |k|, b \cdot |k| \leq 8 \cdot 10^{18}$, every solvable test case will have a solution with $0 \leq a \cdot k, b \cdot k \leq 2 \cdot 10^{18}$. Together the printed moves should move the piece by x and y in the middle of a large enough board. If there are multiple solutions, any small enough one will be accepted. If the piece cannot reach the target, output the line “impossible”. End the output for the case with a blank line. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $0 \leq a, b, x, y \leq 10^{17}$

Sample Input 1

```
2
1 2 1 0
2 2 1 1
```

Sample Output 1

```
Case #1:
1 2 1
1 -2 0
2 1 -1
2 -1 0
-2 1 -1
-1 2 0

Case #2:
impossible
```