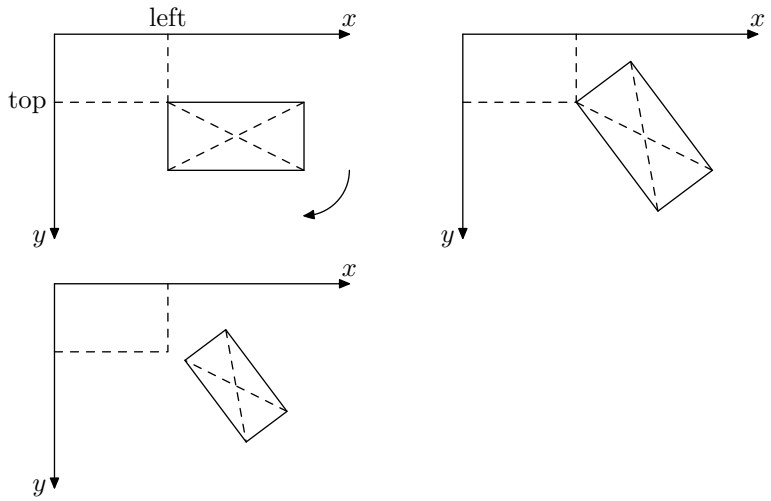
Problem A. Drawing with CSS

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

Vasya is developing a browser online game and he wants to draw some objects in a browser.

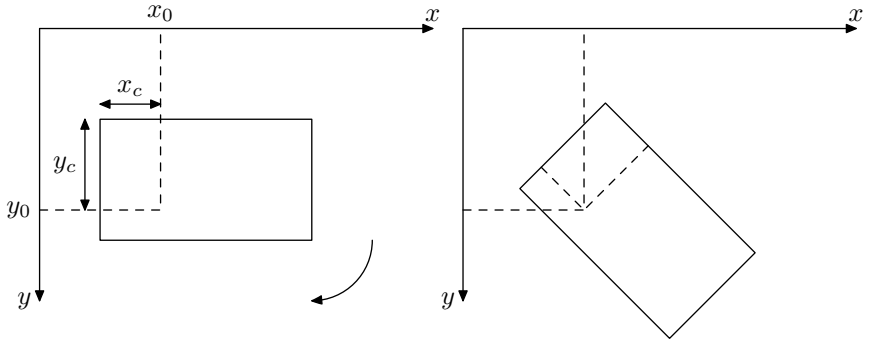
Drawing in a browser uses CSS rules those specify the position of an object, the size, the rotation and the scale. For simplicity in this task, all objects are rectangles. The coordinate system is defined in a browser so that the X-axis goes from the left to the right, and the Y-axis goes from the top to the bottom. A CSS rule specifies the position of a rectangle as a pair of coordinates of the top-left corner $(left, top)$, where $left$ means the X-coordinate, and top means the Y-coordinate. To rotate a rectangle a rule must specify the angle α in degrees to which the object will be rotated about its center in the clockwise direction. CSS rules support scaling of objects as well. To scale an object one should use CSS statement with width/height of the object image.

The order of a browser's actions is the following: the positioning of an rectangle according to the coordinates of the top-left corner, then rotation about the center, and the last action is scaling about the center.



Vasya's rectangles are not so simple, and it turned out that the images in them are such that they shouldn't be rotated and scaled about the center of the rectangle, but about the another point, which we denote as *pivot*.

Vasya defines the pivot as (x_c, y_c) for every rectangle that he has, where x_c is offset by the X-coordinate from the top-left corner, and y_c is offset by Y-coordinate. Now he wants to draw a rectangle so that the pivot gets exactly to the point (x_0, y_0) in the browser. After that, a rectangle is rotated by clockwise angle *alpha* about the pivot and scaled about the pivot by the factor s.



Your task is to generate CSS rule to place rectangles as Vasya wants.

Input

The first line of the input contains integer n ($1 \le n \le 10^4$), the number of rectangles. The following n lines contain descriptions of rectangles, one description per line. A description consists of seven integers w, h, x_c, y_c, x_0, y_0, α ($1 \le w \le 1000$, $1 \le h \le 1000$, $0 \le x_c \le w$, $0 \le y_c \le h$, $0 \le x_0 \le 1000$, $0 \le y_0 \le 1000$, $0 \le \alpha < 360$) and one real number s ($0 < s \le 10$) which is given with exactly one digit after the decimal point, where:

- w means the width of a rectangle, h is the height,
- the point (x_c, y_c) defines the position of the pivot in a rectangle,
- the point (x_0, y_0) defines the position of the pivot in a browser,
- α is the clockwise angle for rotation,
- s is the scale.

Output

Print on a single line a CSS rule which places the rectangle as Vasya needs in the format "`.itemN { width: W; height: H; left: L; top: T; transform: rotate(Adeg); }`", where:

- N means the index of a rectangle (all rectangles are numbered from 1 in the order as they are given in the input);
- W means the width of a rectangle painted in a browser;
- H means the height of a rectangle painted in a browser;
- (L, T) is the position of the top-left corner of a rectangle before rotating;
- A is the clockwise angle for rotating about the center of a rectangle.

Strictly follow the output format. The example have extra line-breaks because of limited table width. The values W, H, L, T, A must be printed with exactly one digit after the decimal point. The value A must satisfy the condition $0 \le A < 360$. Every CSS rule should be printed on exactly one line. All attributes are required and you can't remove any of them.

Examples

stdin
2 200 100 50 75 400 200 45 0.5 200 100 50 75 400 200 90 2.0
stdout
.item1 { width: 100.0px; height: 50.0px; left: 376.5px; top: 183.8px; transform: rotate(45.0deg); } .item2 { width: 400.0px; height: 200.0px; left: 250.0px; top: 200.0px; transform: rotate(90.0deg); }