In a popular manager of windows, the following definition is used to maintain the information of the visible windows in the screen of the computer:

```c
struct Rectangle {
    int x_left, x_right, y_down, y_up;
};
```

Here, the rectangles have the parallel sides in the axes x and y, and x_left, x_right, y_down and y_up are respectively the minimal horizontal coordinate, the maximal horizontal coordinate, the minimal vertical coordinate, and the maximal vertical coordinate of each rectangle.

Write a procedure that reads a rectangle:

```c
void read(Rectangle& r);
```

which is given in the input with the four integer numbers x_left, x_right, y_down and y_up in this order.

Write also a function that indicates the relationship that have two given rectangles r1 and r2:

```c
int relationship(const Rectangle& r1, const Rectangle& r2);
```

that must return 1 if r is inside r2, 2 if r2 is inside r1, 3 if none is inside the other one but the rectangles intersect, 4 if the rectangles are identical, and 0 otherwise (if the rectangles do not have any point in common).

Suppose that two rectangles intersect even if they coincide only in a segment or a point. Moreover, suppose that all the rectangles are correctly formed, that is, that x_left is strictly smaller than x_right, and that y_down is strictly smaller than y_up.

Use these definitions and procedures to write a program that reads a series of pairs of rectangles, and for each one prints which relationship have.

**Input**

Input consists of a natural n, followed by n lines, each one with two rectangles (eight integer numbers).

**Output**

For each pair of rectangles, print their relationship as it is shown in the examples.

<table>
<thead>
<tr>
<th>Sample input</th>
<th>Sample output</th>
</tr>
</thead>
</table>
| 6
2 3 4 6 0 5 2 8
0 10 0 10 9 10 0 1
0 2 0 2 1 3 1 3
-1 1 -2 2 -1 1 -2 2
0 1 0 2 5 7 4 7
0 2 0 2 2 4 2 4 | the first rectangle is inside the second one
the second rectangle is inside the first one
rectangles intersect
rectangles are identical
rectangles do not intersect
rectangles intersect |