

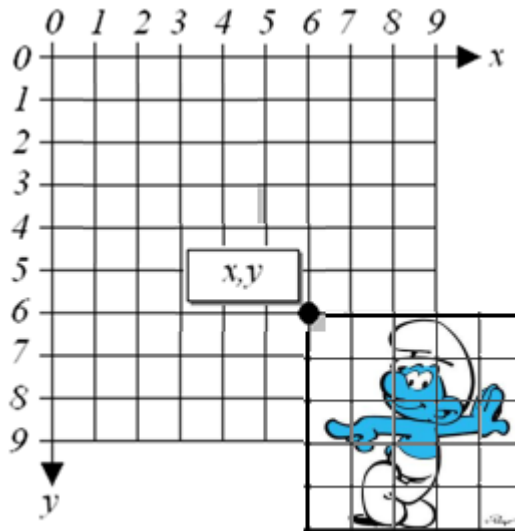
Active Sprites

Filename: sprite

The Problem

In a 2D game it is necessary to update the position of the sprites displayed to the screen every frame. So that you are not updating too many sprites you may want to update a sprite only if it is active, or inside the screen. Your job is to create a function that determines if the sprite is NOT active. You know that the screen is 800x600 pixels, and you are given the position, and dimensions of each sprite. The position (x,y) of the sprite, will correspond to the pixel location of its top left corner. The dimensions of each sprite are given by their width, W (length in x-axis), and height, H (length in the y-axis). The sprite is considered active if any part of the image (at least one square pixel) is on the screen, so the smurf sprite shown below is still active.

Remember that you will be using the pixel coordinate system. This means that the top left corner of the screen is $(0, 0)$, where X increases to the right, and Y increase down. The (x, y) coordinates of the point below are $(6, 6)$.



The Input

The first line of the input file will contain a single positive integer, n , representing the number of input test cases. The following n lines will contain 4 integers. The 4 integers will be the X position, the Y position, the width W , the height H , where X, Y are between -999 and 999 , inclusive and W, H are between 0 and 999 , inclusive. You must determine if the sprite is active or not.

The Output

For each test case, output a single line with the following formats:

```
Case #X: The sprite is active.
Case #X: The sprite is NOT active.
```

where X is the 1-based case number.

Sample Input

```
5
900 700 200 100
700 900 4 5
325 532 218 111
-546 166 250 550
-1 -1 800 800
```

Sample Output

```
Case #1: The sprite is NOT active.
Case #2: The sprite is NOT active.
Case #3: The sprite is active.
Case #4: The sprite is NOT active.
Case #5: The sprite is active.
```