

460 Overlapping Rectangles

When displaying a collection of rectangular windows on a SUN screen, a critical step is determining whether two windows overlap, and, if so, where on the screen the overlapping region lies.

Write a program to perform this function. Your program will accept as input the coordinates of two rectangular windows. If the windows do not overlap, your program should produce a message to that effect. If they do overlap, you should compute the coordinates of the overlapping region (which must itself be a rectangle).

All coordinates are expressed in “pixel numbers”, integer values ranging from 0 to 9999. A rectangle will be described by two pairs of (X, Y) coordinates. The first pair gives the coordinates of the lower left-hand corner (X_{LL}, Y_{LL}) . The second pair gives the coordinates of the upper right-hand coordinates (X_{UR}, Y_{UR}) . You are guaranteed that $X_{LL} < X_{UR}$ and $Y_{LL} < Y_{UR}$.

Input

Input for your program consists of sets of two lines. The first contains the integer numbers X_{LL} , Y_{LL} , X_{UR} and Y_{UR} for the first window. The second contains the same numbers for the second window.

Input is terminated by a four zeros line.

Output

If the two windows do not overlap, print the message “No Overlap”. If the two windows do overlap, print 4 integer numbers giving the X_{LL} , Y_{LL} , X_{UR} and Y_{UR} for the region of overlap. These values should be written in order pair notation (i.e. $(X_{LL}, Y_{LL}) (X_{UR}, Y_{UR})$).

Note that two windows that share a common edge but have no other points in common are considered to have “No Overlap”.

Sample Input

```
0 20 100 120
80 0 500 60
0 0 0 0
```

Sample Output

```
(80,20) (100,60)
```