

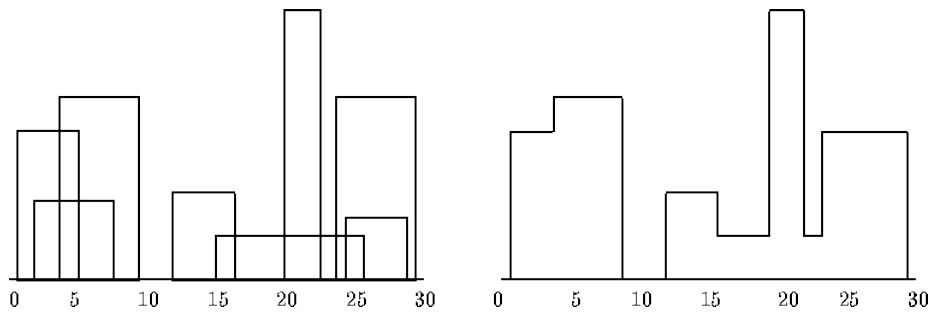
With the advent of high speed graphics workstations, CAD (computer-aided design) and other areas (CAM, VLSI design) have made increasingly effective use of computers. One of the problems with drawing images is the elimination of hidden lines — lines obscured by other parts of a drawing.

You are to design a program to assist an architect in drawing the skyline of a city given the locations of the buildings in the city. To make the problem tractable, all buildings are rectangular in shape and they share a common bottom (the city they are built in is very flat). The city is also viewed as two-dimensional. A building is specified by an ordered triple  $(L_i, H_i, R_i)$  where  $L_i$  and  $R_i$  are left and right coordinates, respectively, of building  $i$  ( $L_i < R_i$ ) and  $H_i$  is the height of the building. In the diagram below buildings are shown on the left with triples

$$(1, 11, 5), (2, 6, 7), (3, 13, 9), (12, 7, 16), (14, 3, 25), (19, 18, 22), (23, 13, 29), (24, 4, 28)$$

the skyline, shown on the right, is represented by the sequence:

$$(1, 11, 3, 13, 9, 0, 12, 7, 16, 3, 19, 18, 22, 3, 23, 13, 29, 0)$$



## Input

The input is a sequence of building triples. All coordinates of buildings are integers less than 10,000 and there will be at least one and at most 5,000 buildings in the input file. Each building triple is on a line by itself in the input file. All integers in a triple are separated by one or more spaces. The triples will be sorted by  $L_i$ , the left  $x$ -coordinate of the building, so the building with the smallest left  $x$ -coordinate is first in the input file.

## Output

The output should consist of the vector that describes the skyline as shown in the example above. In the skyline vector  $(v_1, v_2, v_3, \dots, v_{n-2}, v_{n-1}, v_n)$ , the  $v_i$  such that  $i$  is an even number represent a horizontal line (height). The  $v_i$  such that  $i$  is an odd number represent a vertical line ( $x$ -coordinate). The skyline vector should represent the “path” taken, for example, by a bug starting at the minimum  $x$ -coordinate and traveling horizontally and vertically over all the lines that define the skyline. Thus the last entry in all skyline vectors will be a 0.

## Sample Input

```
1 11 5
2 6 7
3 13 9
12 7 16
14 3 25
19 18 22
23 13 29
24 4 28
```

## Sample Output

```
1 11 3 13 9 0 12 7 16 3 19 18 22 3 23 13 29 0
```