**Monkey and Banana**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 8248    Accepted Submission(s): 4254**

**Problem Description**

A group of researchers are designing an experiment to test the IQ of a monkey. They will hang a banana at the roof of a building, and at the mean time, provide the monkey with some blocks. If the monkey is clever enough, it shall be able to reach the banana by placing one block on the top another to build a tower and climb up to get its favorite food.  
  
The researchers have n types of blocks, and an unlimited supply of blocks of each type. Each type-i block was a rectangular solid with linear dimensions (xi, yi, zi). A block could be reoriented so that any two of its three dimensions determined the dimensions of the base and the other dimension was the height.   
  
They want to make sure that the tallest tower possible by stacking blocks can reach the roof. The problem is that, in building a tower, one block could only be placed on top of another block as long as the two base dimensions of the upper block were both strictly smaller than the corresponding base dimensions of the lower block because there has to be some space for the monkey to step on. This meant, for example, that blocks oriented to have equal-sized bases couldn't be stacked.   
  
Your job is to write a program that determines the height of the tallest tower the monkey can build with a given set of blocks.

**Input**

The input file will contain one or more test cases. The first line of each test case contains an integer n,  
representing the number of different blocks in the following data set. The maximum value for n is 30.  
Each of the next n lines contains three integers representing the values xi, yi and zi.  
Input is terminated by a value of zero (0) for n.

**Output**

For each test case, print one line containing the case number (they are numbered sequentially starting from 1) and the height of the tallest possible tower in the format "Case case: maximum height = height".

**Sample Input**

1

10 20 30

2

6 8 10

5 5 5

7

1 1 1

2 2 2

3 3 3

4 4 4

5 5 5

6 6 6

7 7 7

5

31 41 59

26 53 58

97 93 23

84 62 64

33 83 27

0

**Sample Output**

Case 1: maximum height = 40

Case 2: maximum height = 21

Case 3: maximum height = 28

Case 4: maximum height = 342

**Source**

[University of Ulm Local Contest 1996](http://acm.hdu.edu.cn/search.php?field=problem&key=University+of+Ulm+Local+Contest+1996&source=1&searchmode=source)

**Recommend**

JGShining   |   We have carefully selected several similar problems for you:  [1087](http://acm.hdu.edu.cn/showproblem.php?pid=1087) [1176](http://acm.hdu.edu.cn/showproblem.php?pid=1176) [1159](http://acm.hdu.edu.cn/showproblem.php?pid=1159) [1203](http://acm.hdu.edu.cn/showproblem.php?pid=1203) [1231](http://acm.hdu.edu.cn/showproblem.php?pid=1231)