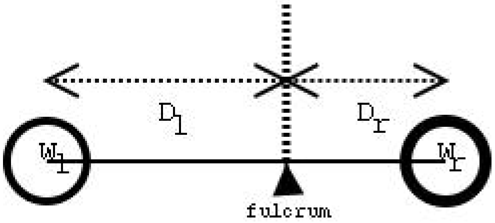
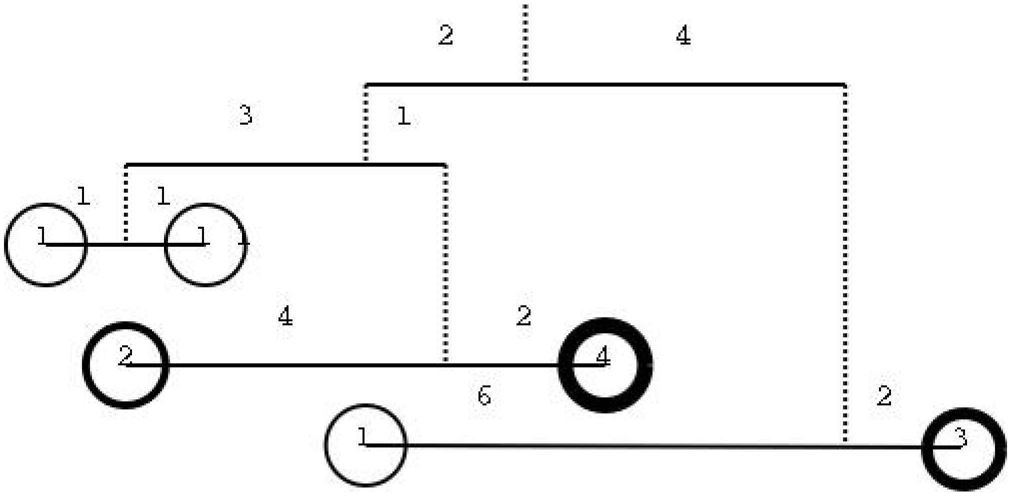
Not so Mobile

Before being an ubiquous communications gadget, a *mobile* was just a structure made of strings and wires suspending colourfull things. This kind of mobile is usually found hanging over cradles of small babies.



The figure illustrates a simple mobile. It is just a wire, suspended by a string, with an object on each side. It can also be seen as a kind of lever with the fulcrum on the point where the string ties the wire. From the lever principle we know that to balance a simple mobile the product of the weight of the objects by their distance to the fulcrum must be equal. That is $\boldsymbol{W}_l \times \boldsymbol{D}_l = \boldsymbol{W}_r \times \boldsymbol{D}_r$ where $\boldsymbol{D}_l$ is the left distance, $\boldsymbol{D}_r$ is the right distance, $\boldsymbol{W}_l$ is the left weight and $\boldsymbol{W}_r$ is the right weight.

In a more complex mobile the object may be replaced by a sub-mobile, as shown in the next figure. In this case it is not so straightforward to check if the mobile is balanced so we need you to write a program that, given a description of a mobile as input, checks whether the mobile is in equilibrium or not.



Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input is composed of several lines, each containing 4 integers separated by a single space. The 4 integers represent the distances of each object to the fulcrum and their weights, in the format: W_l D_l W_r D_r

If W_l or W_r is zero then there is a sub-mobile hanging from that end and the following lines define the the sub-mobile. In this case we compute the weight of the sub-mobile as the sum of weights of all its objects, disregarding the weight of the wires and strings. If both W_l and W_r are zero then the following lines define two sub-mobiles: first the left then the right one.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Write `YES' if the mobile is in equilibrium, write `NO' otherwise.

Sample Input

```
1

0 2 0 4
0 3 0 1
1 1 1 1
2 4 4 2
1 6 3 2
```

Sample Output

```
YES
```

Jose Paulo Leal, ACM-UP'2001