

10089 Repackaging

Coffee cups of three different sizes (identified as size 1, size 2 and size 3 cups) are produced in factories under *ACM* (Association of Cup Makers) and are sold in various packages. Each type of package is identified by three positive integers (S_1, S_2, S_3) where S_i ($1 \leq i \leq 3$) denotes the number of size i cups included in the package. There is no package with $S_1 = S_2 = S_3$.

But recently it has been discovered that there is a great demand for packages containing equal number cups of all three sizes. So, as an emergency step to meet the demand *ACM* has decided to unpack the cups from some of the packages stored in its (unlimited) stock of unsold products and repack them in packages having equal number of cups of all three sizes. For example, suppose *ACM* has the following four types of packages in its stock: $(1, 2, 3)$, $(1, 11, 5)$, $(9, 4, 3)$ and $(2, 3, 2)$. So, one can unpack three $(1, 2, 3)$ packages, one $(9, 4, 3)$ package and two $(2, 3, 2)$ packages and repack the cups to produce sixteen $(1, 1, 1)$ packages. One can even produce eight $(2, 2, 2)$ packages or four $(4, 4, 4)$ packages or two $(8, 8, 8)$ packages or one $(16, 16, 16)$ package or even different combination of packages each containing equal number of size 1, size 2 and size 3 cups. Note that all the unpacked cups are used to produce the new packages, i.e., no unpacked cup is wasted.

ACM has hired you to write a program that will decide whether it is possible to produce packages containing equal number of all three types of cups using all the cups that can be found by unpacking any combination of existing packages in the stock.

Input

The input may contain multiple test cases. Each test case begins with a line containing an integer N ($3 \leq N \leq 1000$) indicating the number of different types of packages that can be found in the stock. Each of the next N lines contains three positive integers denoting respectively the number of size 1, size 2 and size 3 cups in a package. No two packages in a test case will have the same specification.

A test case containing a zero for N in the first line terminates the input.

Output

For each test case in the input print a line containing 'Yes' if packages can be produced as desired, print 'No' otherwise.

Sample Input

```
4
1 2 3
1 11 5
9 4 3
2 3 2
4
1 3 3
1 11 5
9 4 3
2 3 2
0
```

Sample Output

Yes

No