Problema número 665

SemVer++

Tiempo máximo: 1,000 s Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=665

El *versionado semántico* (SemVer, por *Semantic Versioning*) es un esquema de numeración de versiones de software, muy utilizado especialmente en bibliotecas por reducir la pesadilla de las dependencias (*dependency hell*). El número de versión tiene relación directa con la *compatibilidad del API*, de modo que es fácil saber, de manera automática, si una determinada aplicación que depende de una versión concreta de una biblioteca podrá funcionar con una versión posterior. El número de versión tiene por tanto un *significado*, de ahí el nombre de *versionado semántico*.

Bajo este esquema, la versión de una biblioteca se compone de tres números separados por punto, cada uno con un significado, que normalmente se representan como *Major.Minor.Patch*. Así, por ejemplo, la versión 3.1.4 tiene como versión principal la 3, subversión 1 y revisión (o parche) 4.

Un incremento en el número principal de la versión (*major*) indica *un cambio incompatible de API*. Eso significa que, por ejemplo, una aplicación que utilice la versión 3.1.4 de una biblioteca no podrá ser compilada con la versión 4.0.0 porque la forma en la que funciona ha cambiado completamente.

Un incremento en el número de subversión (*minor*) indica una mejora de la librería, ampliando su funcionalidad pero no cambiando nada de lo que ya tuviera la versión previa. Por su parte, un cambio en el número de revisión (*patch*) indica un arreglo de algún error descubierto de la versión previa, pero sin cambios en la funcionalidad.

El problema del versionado semántico es que es imposible saber qué versión seguirá a una dada, porque podría cambiar cualquiera de los tres números. Eso sí, al menos sabemos que cuando un número cambia, los siguientes se reinician a 0.

Entrada

La entrada comienza con un primer número indicando cuántos casos de prueba deberán ser procesados. Cada uno está compuesto por dos números de versión de una determinada biblioteca software, en formato *SemVer*. Todos los números estarán entre 0 y 1000.

Salida

Por cada caso de prueba el programa escribirá "`SI`" si el segundo número de versión *podría ser* el siguiente al primero, y "`NO`" en caso contrario.

Entrada de ejemplo

```
3
3.1.4 3.1.6
0.0.10 1.0.0
1.2.3 1.3.1
```

Salida de ejemplo

```
NO
SI
NO
```

Autores: Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

Revisor: Alberto Verdejo.