

Cuadrados imperfectos

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

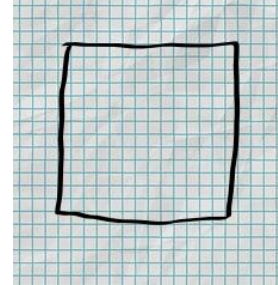
<http://www.aceptaelreto.com/problem/statement.php?id=377>

Cuando se trata de realizar operaciones rápidamente, todo vale en informática. Los programadores buscan los trucos más extraños (al menos vistos desde fuera) para ahorrarse unos pocos ciclos de CPU. Y, egoístamente, hacen bien. Si con eso consiguen mejor respuesta ante las pulsaciones de botones de un editor de imágenes, o mejor realismo en un videojuego, bienvenidos sean.

Una de las optimizaciones más extrañas que se hizo famosa a principios del siglo XXI pertenecía, precisamente, al mundo de los videojuegos. Una operación que se necesitaba realizar muchas veces por segundo era la inversa de la raíz cuadrada de un número, es decir $y = 1/\sqrt{x}$.

El truco para ahorrarse el cálculo, sacrificando mínimamente la precisión, consistía en interpretar el valor original de x no como un número real (`float`) si no como si fuera un entero de 32 bits (`int`) y utilizando restas, desplazamientos (en aritmética entera) y... el “número mágico” `0x5f3759df` se conseguía un entero que tenía casi la misma representación binaria que el `float` del resultado buscado.

Una optimización mucho más fácil de entender y demostrar nos permite afirmar con rotundidad si un número entero dado *no* es un cuadrado perfecto. Y se basa en que todos los cuadrados perfectos, cuando se representan en hexadecimal (o base 16), terminan en 0, 1, 4 o 9. Eso significa que, dado el número en esa base, si no acaba en uno de esos dígitos, podemos asegurar que el número no tiene raíz cuadrada entera; eso sí, lo contrario no es cierto; si el número acaba en alguno de esos dígitos, podría ser cuadrado perfecto o no.



Entrada

La entrada está formada por distintos casos de prueba, cada uno en una línea.
Cada caso de prueba es un número (en base 10) de hasta 1.000 dígitos.

Salida

Para cada caso de prueba se escribirá “IMPERFECTO” si, gracias a la comprobación indicada, es seguro que el número no es un cuadrado perfecto y “NO SE” si la prueba no permite asegurarlo.

Entrada de ejemplo

```
4
14
9
19
```

Salida de ejemplo

```
NO SE
IMPERFECTO
NO SE
IMPERFECTO
```

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