Problema - Inverso de divisores

Se dice que un número M divide a un número N si el residuo de N/M es igual a 0, un número tiene una cierta cantidad de divisores, por ejemplo 24 tiene 8 divisores:

```
24 / 1 = 24 (residuo 0)
24 / 2 = 12 (residuo 0)
24 / 3 = 8 (residuo 0)
24 / 4 = 6 (residuo 0)
24 / 6 = 4 (residuo 0)
24 / 8 = 3 (residuo 0)
24 / 12 = 2 (residuo 0)
24 / 24 = 1 (residuo 0)
```

En este problema se te dará el número de divisores que tiene cierto número y tu debes hallar el mínimo número que tengan esos divisores.

Entrada

Se te darán varios números que representan el número de divisores que debe tener algún número, este número de divisores estará entre 1 y 10^9. La entrada termina cuando no hay mas datos.

Salida

Para cada número que recibas en la entrada imprimir en una línea, el mínimo número posible que tenga esa cantidad de divisores de no existir o que este número sea mayor que 10^9 imprimir -1.

Ejemplos de entrada	**Ejemplos de salida**
8 64	24 7560

Tarea 1: 30 puntos

La entrada son números entre 1 y 10.

Tarea 2: 70 puntos

La entrada son números entre 1 y 10^9.

Nota: El primer caso del ejemplo corresponde a la tarea 1, el segundo caso del ejemplo corresponde a la tarea 2.

Problema - Inverso factorial

Se define al factorial de un número N, como el producto de todos los números entre 1 a N: `N! = 1 * 2 * 3 * 4 * ... * N` con $0! = 1$.

Por ejemplo `5! = 1 * 2 * 3 * 4 * 5 = 120`

Lo que se pide en este problema, como el título dice, es calcular el inverso del factorial, en otras palabras encontrar un número tal que aplicando el factorial es igual al número de la entrada.

Por ejemplo: el inverso de 120 es 5, ya que $5! = 120$, el inverso de 6 es 3 ya que $3! = 6$

Entrada

Consta de varios números que tienen entre 1 y 10^6 dígitos, estos son factoriales válidos. La entrada termina cuando no existen más datos. Los números son mayores a 1.

Salida

Para cada número que recibas en la entrada le corresponde una línea en la salida, que es el cálculo del inverso del factorial.

Ejemplos de entrada	**Ejemplos de salida**
120 6	5 3

Tarea 1: 40 puntos

La entrada son números entre 1 y 18 dígitos.

Tarea 2: 60 puntos

La entrada son números entre 1 y 10^6 dígitos.

Sugerencias:

1. Usar lectura rápida, si no sabes que es eso agrega esto a tu código:

```
ios::sync_with_stdio(0);
```

```
cin.tie(0);
```

en la primera línea del main.

Ejemplo:

```
#include <iostream>
using namespace std;
int main(){
ios::sync_with_stdio(0);
cin.tie(0);
```

2. No usar *endl* como salto de línea, usar "`\n`" en vez de esto.

3. Recuerda leer hasta fin de archivo son varios números por caso de entrada. Esto es intencionado.

Problema - Competidores OBI

A la Fase Nacional de la Olimpiada Científica Estudiantil, han llegado estudiantes de muchos departamentos de Bolivia.

Todos los estudiantes tienen una credencial y en ella un código único, y por alguna extraña razón el código de los competidores del área de informática tienen una particularidad: Tienen al menos 3 números primos de un dígito y la suma de estos números primos es par.

Mishell, una competidora del área de matemáticas se ha dado cuenta de ello y por eso va contando competidores en el área de informática en cada grupo que ella conoce, ella anota y hace sus cálculos de manera manual ya que no sabe programar. Por ello te pide ayuda, ya que vio tu credencial.

Realiza un programa donde ingresas n números que representan los códigos de las credenciales y cuente los competidores en el área de informática.

Entrada

La primera línea tendrá un número $n < 4000$, que es la cantidad de casos de prueba. Seguido de n líneas. En cada línea se tiene un número $x > 1$ y $x < 100000000$. Cada x corresponde al código de una credencial.

Salida

Imprimir sólo un número, el cual indica la cantidad de competidores en el área de informática.

Ejemplos de entrada	**Ejemplos de salida**
3 12341689 567912 23456	2

Problema - Robo

Son tiempos difíciles en el país de las hadas, ahora existe escasez del polvo mágico que usan para volar y otras capacidades que las hadas tienen. La prqueña TinTan ha comenzado a sospechar que hay una ladrona entre ellas!

Desde que comenzó la escasez el Consejo de Hadas decidió entregar diariamente el polvo mágico en forma de cubo, lamentablemente los duendes que procesan el polvo no se han puesto de acuerdo en las dimensiones y entregan a cada hada un cubo de dimensiones distintas, eso si, por justicia entregan la misma cantidad a todas las hadas.

Desde hace varios días, algunas compañeras de TinTan se han quejado que no les alcanz'o el polvo mágico para sus actividades, TinTan sabe que varias de ellas son muy responsables y no malgastarían el preciado polvo; solo queda la posibilidad de una ladrona! será posible un hada ladrona?

Cada día solamente una hadita se queja de falta de polvo, por esta razón TinTan sabe que solamente existe una ladrona.

TinTan ha acudido al programador Tu, si ... Tu como lo oyes, Tu debes ayudarle.

Para la investigación Tu tiene una lista entregada por TinTan que provee el nombre de cada hadita que recibe polvo diario y las dimensiones del cubito (largo, ancho, y alto).

Entrada

La entrada consiste un valor D $(1 < D < 100)$en la primera línea que representa el número de días para los que se investiga, luego para cada d'ia se tiene un valor N para el número de haditas que recibieron polvo mágico en ese día con $1 < N < 1000$, en las siguientes n líneas se tiene el nombre de la hadita (máximo 25 caracteres, como verás las hadas tienen nombres exóticos difícles de pronunciar para los mortales simples) y las dimensiones del cubo: largo, ancho y alto; estas dimensiones son un valor en el rango $[1, 250]$

Los datos de la entrada representan el volumen actual de los cubos.

Salida

Imprimir, por cada día (caso de prueba), una línea con el nombre de la ladrona, el nombre de la víctima y la cantidad de polvo robado siguiendo el patrón siguiente: **X ha robado a Y un volumen de M**, siendo X, Y nombres de hadas y M la cantidad robada.

En caso de no haber ladrona en un día se debe mostrar como resultado una línea con el mensaje **No hubo robo hoy**

Ejemplos de entrada	Ejemplos de salida
3 4 anhana 2 2 2 morgana 1 4 2 calipso 2 3 1 helga 2 5 1 4 anhana 2 2 2 morgana 2 2 2 calipso 2 2 2 helga 2 2 2 5 anhana 2 2 1 morgana 2 2 2 calipso 2 4 2 helga 2 2 2 carlin 2 1 2	helga ha robado a calipso un volumen de 2 No hubo robo hoy calipso ha robado a anhana un volumen de 4 calipso ha robado a carlin un volumen de 4

Tarea 1 (30%)

Se debe pensar soluciones cuando roba a una sola hada. Vea el ejemplo de los dos primeros casos.

Tarea 2 (70%)

Se debe pensar soluciones cuando roba a varias hadas. Vea el tercer ejemplo.

Problema - $1/998001 =?$

IMPORTANTE: Responde en el cuadro habilitado como corresponda a cada uno de los casos planteados en la sección Problema. Escribe una línea por cada respuesta, una respuesta por caso; no todas las respuestas en la misma línea, no todas las respuestas separadas con espacios. UNA respuesta por cada linea en el cuadro habilitado.

Una relación importante de las matemáticas es 1/998001, que genera un resultado de precisión infinita, lo curioso es que es una serie del 0 al 999, escrito en tres cifras, una vez que termina, inicia de nuevo la serie:

```
1/998001 = 0.000001002003004005006007008009010011012013014015016017018019020021
022023024025026027028029030031032033034035036037038039040041042043044045046047048
049050051052053054055056057058059060061062063064065066067068069070071072073074075
076077078079080081082083084085086087088089090091092093094095096097098099100101102
103104105106107108109110111112113114115116117118119120121122123124125126127128129
130131132133134135136137138139140141142143144145146147148149150151152153154155156
157158159160161162163164165166167168169170171172173174175176177178179180181182183
184185186187188189190191192193194195196197198199200201202203204205206207208209210
211212213214215216217218219220221222223224225.........
```

Cálculus les ha pedido a sus estudiantes que resuelvan el problema sin usar una calculadora por lo que deben hacer el trabajo manualmente y entregarle sus respuestas. Cálculus ya no se preocupa cómo encontrar el resultado, sino más bien en verificar que el resultado sea correcto. Es así que su tarea es cerciorarse de que los términos de la relación se encuentren en las propuestas de resultado(truncadas) que le generan sus estudiantes, en caso de que no sea así su nota dependerá de la cantidad de términos faltantes y la precisión de la propuesta. Por lo que necesita saber qué términos no se encuentran en la respuesta si es incorrecta, caso contrario si la respuesta es correcta debe emitir un informe de "CORRECTO".

Por ejemplo, si uno de sus estudiantes le entrega la siguiente respuesta:

`0.000001002003004005006007008009010011012013014015016017018020021022023`, debera indicar los términos que faltan, en este caso particular falta el `019`.

Tu tarea es identificar si una propuesta de solución a la división propuesta es correcta, en este caso debes emitir "CORRECTO", en caso de que no sea hay que indicar los términos que faltan.

Entrada

La entrada consiste de un número entero n que indica en número de casos de prueba, le siguen n líneas y cada una de ellas es una propuesta de solución s que representa a un número real de la forma $0.ddddddd...$, la cantidad de cifras de s incluido el . no excede a 20000 caracteres y no es menor que 3.

Salida

La salida por cada caso de prueba se muestra en una línea, la misma que contiene la cadena "CORRECTO" o una lista de términos faltantes separados por un espacio en blanco.

Ejemplos de entrada

```
5
0.000001002003004005006007008009010011012013014015016017018020021022023
0.000001003004005007
0.00000100200300400500600700800901
0.00001
0.0000010
```

Ejemplos de salida

```
019
002 006
CORRECTO
001 002 003 004 005 006 007 008 009
CORRECTO
```

Tarea 1: 50 puntos

Los casos de entrada tendrán cantidad de dígitos después del punto decimal equivalente a un m'ultiplo de 3. Por ejemplo:

```
0.000010
```

Tarea 1: 50 puntos

Los casos de entrada tendr'an cualquier cantidad de dígitos. Por ejemplo:

```
0.00000101
```