Programa-Me 2012
Regional Madrid
Problemas

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN

UNIÓN EUROPEA
Fondo Social Europeo
Invierte en tu futuro

Patrocinado por

Universidad Complutense Madrid
Vicerrectorado de Informatica y Comunicaciones

Facultad de Informática

CORITEL
An Accenture Company

StudyPLAN Tecnología Educación

eBeam

El Corte Inglés

Realizado en el IES Antonio de Nebrija. Móstoles

CONSEJERÍA DE EDUCACIÓN
Comunidad de Madrid

Listado de problemas

A Apuesta con recetas **3**

B Escudos del ejército romano **5**

C Constante mágica **7**

D Chicles de regalo **9**

E Avituallamiento en las etapas ciclistas **11**

F Conjugar verbos **13**

G ¿Cuántas me llevo? **15**

H Números vampiro **17**

I Divisores del factorial **19**

J Una, dola, tela, catola... **21**

A
Apuesta con recetas

Pedro ha recibido el siguiente correo:

> “
> Has sido invitad@ a un intercambio de recetas.
> Espero que participes, escogí a quien me pareció que se iba a divertir con esto.
>
> 1. Por favor, envía una receta a la persona cuyo nombre figura como número 1 aquí abajo (aunque no la conozcas). Debe ser algo rápido, fácil y con ingredientes corrientes. De hecho, la mejor receta es aquella que conoces de memoria y que eres capaz de escribir y enviar de inmediato. No te preocupes mucho y envía aquella que haces cuando tienes poco tiempo para entretenerte.
> 2. Después de enviar la receta a la persona con el número 1 de aquí abajo, y sólo a esa persona, copia esta carta en un nuevo correo, coloca mi nombre en la primera posición y pon el tuyo en el número 2. Sólo deben aparecer tu nombre y el mío cuando envíes tu mensaje.
> 3. Envía esto a 8 amigos. Si no consigues hacerlo en 5 días, comunícamelo para ser justo con los participantes.
>
> Deberías recibir muchas recetas. ¡Es divertido ver de dónde vienen!
> Raramente las personas desisten ya que todos agradecemos nuevas ideas. El retorno es rápido ya que únicamente hay 2 nombres en la lista y cada uno sólo lo tiene que hacer una vez.
> POSICIÓN 1 : javier@acmicpc.org
> POSICIÓN 2 : luis@swerc.eu
> ”

Pedro se lo cuenta a sus amigos Pilar y Marco un día tomando café y, casualmente, ellos también han recibido el mismo mensaje. Tienen curiosidad por saber cuántas recetas podrían recibir si envían los correos a las 8 personas que dicta el mensaje.

No se ponen de acuerdo en si les llegarían muchas o pocas recetas, de modo que deciden hacer una apuesta al respecto. Cada uno debe aventurar cual será la *media* de recetas recibidas por los tres. Para que no haya ventaja por parte de ninguno de los amigos, cada uno de ellos escribirá en un papel, de forma secreta, su apuesta. Cuando se han anotado las tres, se hacen públicas. Al cabo de un mes realizarán un recuento de las recetas recibidas y calcularán la media. Aquél cuya apuesta se aleje más de dicha media tendrá que hacer una cena a los otros dos con algunas de las recetas conseguidas.

En cuanto lo ha pensado un poco, nuestro amigo Pedro se ha dado cuenta de una cosa: en este juego ganar no sirve de nada, porque no hay premio; lo importante es *no perder*, para no tener que cocinar. A Pedro le pone nervioso tener que cocinar para otros, así que, a la vista de las apuestas, quiere evitar perder a toda costa.

Afortunadamente, puede hacer cómplice a muchos amigos (y amigos de amigos) para que le ayuden a recibir, de acuerdo a las reglas del mensaje, el número de recetas que él les pida. Para evitar que Pilar y Marco noten la trampa, les quiere pedir un número de recetas que garantice que no pierde nunca (independientemente de las recetas recibidas por Pilar y Marco), pero que *le deje lo más cerca posible de la media*, sin perder, para no despertar sospechas.

Los tres amigos son lo suficientemente avispados para no realizar apuestas imposibles, por encima del límite que les impone el número de amigos a los que envían el correo.

Realiza un programa que ayude a Pedro a solucionar su problema.

Entrada

Como entrada se recibirán múltiples casos de prueba. Cada uno estará compuesto de los *tres valores enteros mayores o iguales que 0* que forman la apuesta. El primero de ellos será el número de recetas apostado por Pilar, el segundo por Marco y el último por Pedro. La entrada finalizará cuando la triada comience por un valor negativo.

Salida

Para cada caso de prueba, el programa escribirá el número de recetas que deberá recibir Pedro para, sin levantar sospechas, no perder. Si es imposible que Pedro pierda, o el juego se considera nulo porque hay coincidencia en las tres apuestas, no molestará a ninguno de sus amigos, por lo que el programa deberá escribir `0` (es decir, un *cero*). Si no existe un número de recetas que permita no perder nunca a Pedro, se mostrará la letra `I`. Se tendrá en cuenta que un empate entre dos de los jugadores significará perder, pues ambos tendrán que cocinar para el tercero. Así, por ejemplo, si otro participante hace la misma apuesta que Pedro, éste tendrá que intentar que pierda el participante que ha introducido la apuesta distinta.

Entrada de ejemplo

```
5 17 32
5 32 17
31 17 5
60 55 50
-1 -1 -1
```

Salida de ejemplo

```
56
0
I
36
```

Fuente

Basado en una historia real.

B
Escudos del ejército romano

Son famosas las formaciones que el antiguo ejército romano utilizaba para entrar en batalla. En esas formaciones, los legionarios se agrupaban en una figura geométrica (normalmente un rectángulo) y protegían tanto los flancos como la parte superior utilizando escudos. Los legionarios que ocupaban posiciones interiores cubrían la parte superior colocando el escudo sobre su cabeza, mientras que los que ocupaban los flancos llevaban dos y hasta tres escudos: uno para proteger la parte superior y uno o dos escudos (si estaban en la esquina) para proteger los laterales. Con esta formación, todos los legionarios quedaban protegidos por los escudos y eran muy difíciles de vencer.

Cuenta la historia[1] que existió un general que estableció que la mejor figura para la formación no era la rectangular sino la cuadrada, de forma que el número de filas y columnas de legionarios coincidía. El problema al que se enfrentaba este general era decidir en cuántas formaciones (y de qué tamaño) debía separar su ejército para que:

- No quedara ningún legionario fuera de una formación (aunque admitía formaciones de *un único legionario*[2]).
- Se minimizara el número de escudos necesarios para protegerlos.

Nuestro general, después de hacer muchos cálculos, decidió que la mejor manera de que estas dos condiciones se cumpliesen era comenzar haciendo el cuadrado más grande posible con sus legionarios. Con los que le quedasen libres volvía a repetir la operación, y así hasta que no quedasen legionarios que formar[3].

Por ejemplo, si el número de legionarios en el ejército era 35, la manera utilizada por el general para hacer la formación consistía en un cuadrado de 25 legionarios (5x5), otro de 9 (3x3) y otro de 1 (1x1):

```
* * * * *
* * * * *   * * *
* * * * *   * * *   *
* * * * *   * * *
* * * * *
```

Esta formación requería un total de 71 escudos.

Entrada

La entrada estará compuesta de múltiples casos de prueba, cada uno en una línea.

Cada caso de prueba indicará el número de legionarios en el ejército que se quiere poner en formación (un ejército tiene como mucho diez millones de legionarios). La entrada terminará con un ejército de cero legionarios, que no provocará salida.

Salida

Para cada caso de prueba se escribirá una línea que indicará el número de escudos mínimo que necesitamos para cumplir las restricciones del general.

[1] Bueno, la historia inventada por los autores de este problema...

[2] No es de extrañar que ningún legionario quisiera quedarse solo, ¡tenía que acarrear un montón de escudos para estar protegido!

[3] En una ocasión un legionario raso de su ejército le mostró que su teoría no era cierta. El pobre legionario terminó devorado por los leones, y su demostración se perdió en la noche de los tiempos.

Entrada de ejemplo

```
35
20
10
0
```

Salida de ejemplo

```
71
44
26
```

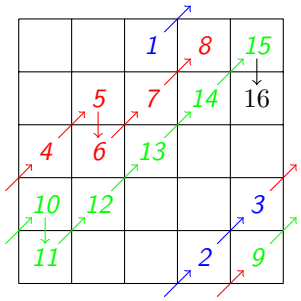
C
Constante mágica

Una matriz de NxN números se considera cuadrado mágico si la suma de cada una de sus filas, cada una de sus columnas y sus diagonales principal y secundaria tienen el mismo valor (la llamada *constante mágica*).

Aunque la generación de cuadrados mágicos no siempre es fácil, cuando tienen tamaño impar existen varios métodos sencillos de generarlos. Uno muy conocido es el llamado *método siamés*, que debemos a Simon de La Loubère quién lo publicó en 1693. Según este método se empieza con un entero cualquiera (el 1 por ejemplo) que se coloca en el cuadrado central de la fila superior. Después se van rellenando los cuadrados en diagonal (hacia arriba y la derecha) con números consecutivos al inicial, teniendo en cuenta que:

- Si al avanzar "salimos fuera" del cuadrado por la parte superior en la columna `i`, se entra por la parte inferior en la columna `i+1`
- Si al avanzar "salimos fuera" del cuadrado por la derecha en la fila `j`, se entra por la izquierda en la fila `j-1`.
- Si la siguiente posición está ocupada colocamos el número justo debajo de la posición donde colocamos el último número. Si estamos en la última fila, volvemos a la primera.

La manera de conseguir el cuadrado mágico de orden 5 con este procedimiento es el que se detalla en la siguiente figura:



Este cuadrado mágico, una vez completo, tendrá 65 como *constante mágica*.

Entrada

La entrada estará compuesta de distintos casos de prueba. Cada caso de prueba consiste en una línea con dos números: n y k. El primero de ellos, n, indica el tamaño de cada lado del cuadrado y será siempre un número impar mayor o igual que 1 y menor que 100. El segundo, k, indica el número que aparecerá en el centro de la fila superior y será siempre mayor o igual que 0 y menor que 1000.

Los casos de entrada finalizan con el caso especial "`0 0`", que no debe generar salida.

Salida

Para cada caso de prueba, se mostrará la *constante mágica* del cuadrado resultante.

Entrada de ejemplo

```
5 1
3 0
3 4
0 0
```

Salida de ejemplo

```
65
12
24
```

Fuente

http://en.wikipedia.org/wiki/Siamese_method

D
Chicles de regalo

Para tener contentos a los niños y de paso intentar que no tiren al suelo los envoltorios de los chicles, la empresa BubbaGum ha decidido dar chicles gratis a aquellos que lleven los envoltorios de otros chicles ya consumidos.

Ha empezado regalando un chicle por cada cinco envoltorios. Eso significa que si compramos 25 chicles podremos en realidad disfrutar de 31: al comernos los 25 comprados tendremos 5 chicles más gratis con los que conseguimos otros 5 envoltorios adicionales con los que podemos ir a la tienda a por otro más. Si la estrategia funciona, la empresa probará a realizar otras ofertas.

La pregunta que nos hacemos es ¿cuántos chicles nos podremos comer según la oferta que haya en ese momento?

Entrada

La entrada estará compuesta de una serie de casos de prueba. Cada caso de prueba está compuesto de una única línea que tiene tres números enteros (no mayores de 10^9) separados por espacios. Los dos primeros números indican cuántos envoltorios se necesitan para que nos den qué cantidad de chicles. El tercer número indica el número de chicles que compramos inicialmente. Ninguno de esos números será negativo; el primero, además, será siempre mayor que cero. Los casos de prueba terminan cuando *los tres* números son cero.

Salida

Para cada caso de prueba se escribirá una línea que tendrá, a su vez, dos números. El primero indicará el número total de chicles que nos comemos, y el segundo con cuántos envoltorios nos quedamos al final.

Si la oferta hace que la empresa se arruine con nosotros se escribirá `RUINA`. Se considerará que la empresa va a la ruina cuando tenga que regalarnos chicles de forma indefinida.

Entrada de ejemplo

```
5 1 25
5 1 5
10 1 100
2 5 20
0 0 0
```

Salida de ejemplo

```
31 1
6 1
111 1
RUINA
```

E
Avituallamiento en las etapas ciclistas

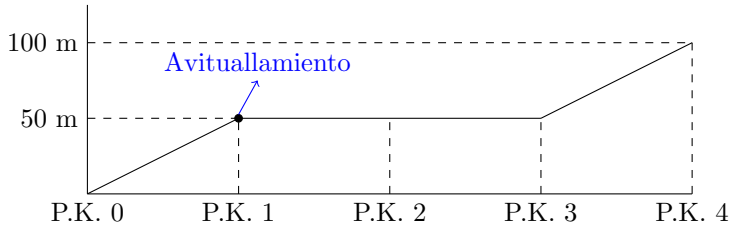
En toda etapa ciclista que se precie hay un momento en el que se permite a los participantes recuperar fuerzas recogiendo comida que voluntarios apostados en los márgenes de la carretera les dan dentro de unas bolsas.

La organización de esas etapas debe decidir en qué punto kilométrico colocan la zona de avituallamiento (el lugar en el que se les proporciona las bolsas). El sentido común indica que:

- Debe ser en una zona llana (si es subida los ciclistas estarán más concentrados en pedalear que en comer; si es bajada estarán atentos a no caerse con el incremento de velocidad...).
- La zona llana debe ser la más larga que haya en toda la etapa, para darles tiempo a comer con calma.

Para tomar esta decisión, te han pedido ayuda. Ellos te darán la altura sobre el nivel del mar a la que empieza cada uno de los kilómetros de la etapa, y tendrás que decidir en qué punto kilométrico se deberán colocar los encargados del avituallamiento.

Como ejemplo, supongamos una etapa de 4 kilómetros, en la que la salida, situada en el punto kilométrico (P.K.) 0, esté sobre el nivel del mar, los P.P.K.K. 1, 2 y 3 estén a una altura de 50 metros y, por último, la meta (P.K. 4), esté a 100 metros de altitud. Con esta configuración de etapa, existen dos kilómetros llanos (que comienzan en los P.P.K.K. 1 y 2), por lo que el avituallamiento se colocará en el P.K. 1, para que los corredores tengan dos kilómetros completos para comer.



Entrada

La entrada está compuesta de múltiples etapas, cada una en una línea. Una etapa está formada por una secuencia de al menos dos números mayores o iguales que cero separados por espacios. Cada uno de esos números representa la altura sobre el nivel del mar al principio del kilómetro, siendo el primer número la altura al principio del primer kilómetro (P.K. 0), el segundo al principio del segundo kilómetro, etc. Cada etapa termina con un `-1` que no debe tenerse en cuenta en el cálculo. La última altura válida leída es la altura a la que se encuentra la meta.

La entrada termina con una "etapa vacía", es decir una línea que contiene únicamente un `-1`.

Salida

Para cada etapa se escribirá una línea donde aparecerán dos números separados por un espacio: el punto kilométrico donde colocar el punto de avituallamiento y el número de kilómetros llanos que tienen por delante.

Ten en cuenta que:

- El principio de la etapa se considera el kilómetro cero.
- Si hay más de una zona candidata, se elegirá la que ocurra antes en la etapa.

- Si en la etapa no hay ninguna zona llana, se escribirá "`HOY NO COMEN`".

Entrada de ejemplo

```
0 50 50 50 100 -1
10 10 -1
0 5 -1
0 50 50 100 100 -1
0 50 50 3 3 3 -1
-1
```

Salida de ejemplo

```
1 2
0 1
HOY NO COMEN
1 1
3 2
```

F
Conjugar verbos

¿Cuántas veces has vivido con pavor los exámenes de lengua en los que te daban un infinitivo y un tiempo verbal y tenías que conjugarlo? Pensabas que no era de ninguna utilidad y que los profes de lengua te obligaban a estudiarlos sólo por hacerte sufrir.

Pues se acabó, ya no hay que estudiar más. Vas a construir un programa que dado un verbo y su tiempo verbal, te lo conjugue. Para ello, cuentas con un ejemplo de cada una de las conjugaciones de los tiempos verbales que generará el programa:

Verbo	Presente (`A`)	Pretérito Perfecto Simple (`P`)	Futuro (`F`)
Saltar	Yo salto Tú saltas Él salta Nosotros saltamos Vosotros saltáis Ellos saltan	Yo salté Tú saltaste Él saltó Nosotros saltamos Vosotros saltásteis Ellos saltaron	Yo saltaré Tú saltarás Él saltará Nosotros saltaremos Vosotros saltaréis Ellos saltarán
Comer	Yo como Tú comes Él come Nosotros comemos Vosotros coméis Ellos comen	Yo comí Tú comiste Él comió Nosotros comimos Vosotros comísteis Ellos comieron	Yo comeré Tú comerás Él comerá Nosotros comeremos Vosotros comeréis Ellos comerán
Vivir	Yo vivo Tú vives Él vive Nosotros vivimos Vosotros vivís Ellos viven	Yo viví Tú viviste Él vivió Nosotros vivimos Vosotros vivísteis Ellos vivieron	Yo viviré Tú vivirás Él vivirá Nosotros viviremos Vosotros viviréis Ellos vivirán

Nota: Nuestro programa *no* pondrá tildes, a pesar de constituir una falta de ortografía.

Entrada

La entrada consistirá en una sucesión de casos de prueba. Cada uno estará compuesto por dos valores, el primero el verbo a conjugar (será siempre regular, de no más de 20 letras y con las dos últimas en minúscula) y el segundo el tiempo verbal en el que queremos conjugarlo. Los posibles tiempos verbales son `A` para el Presente, `P` para el Preterito Perfecto Simple y `F` para el Futuro.

La entrada finalizará con un verbo cualquiera y el carácter `T`, que indicará que el programa debe terminar.

Salida

Para cada caso de prueba el programa mostrará la conjugación del verbo dado, en el tiempo verbal pedido, para cada una de sus personas. Las mayúsculas del verbo original deben conservarse, aunque la salida parezca incorrecta.

Entrada de ejemplo

```
saltar P
Comer A
viVir F
terminar T
```

Salida de ejemplo

```
yo salte
tu saltaste
el salto
nosotros saltamos
vosotros saltasteis
ellos saltaron
yo Como
tu Comes
el Come
nosotros Comemos
vosotros Comeis
ellos Comen
yo viVire
tu viViras
el viVira
nosotros viViremos
vosotros viVireis
ellos viViran
```

G

¿Cuántas me llevo?

Cuando aprendemos a sumar números pronto nos cuentan aquello de "llevarse una": cuando los dos dígitos que sumamos llegan a la decena tenemos "acarreo" que debemos sumar a los siguientes dígitos (de la izquierda).

Cuando nuestros maestros nos ponían ejercicios, antes tenían que contar cuántas veces tendríamos que "llevarnos una" y en base a eso medían la dificultad del ejercicio.

¿Puedes hacer un programa que automatice esa tarea?

Entrada

La entrada estará compuesta de numerosos casos de prueba, cada uno de ellos en una línea. En cada línea apareceran dos números positivos separados por un espacio. Se garantiza que los números no tendrán más de 1000 dígitos.

Los casos de prueba terminan con el caso especial `0 0`, que no provocará salida.

Salida

Para cada caso de prueba hay que escribir en una línea el número de veces que hay "acarreo" en la suma.

Entrada de ejemplo

```
123 456
555 555
123 594
0 0
```

Salida de ejemplo

```
0
3
1
```

Fuente

Basado en el problema "Aritmética primaria" del Juez on-line de la Universidad de Valladolid (código de ejercicio 10035, `http://uva.onlinejudge.org/external/100/10035.html`).

H
Números vampiro

En 1994, Clifford A. Pickover puso de manifiesto la existencia de los temidos *números vampiro*. Los números vampiro sobreviven ocultos entre el resto de nuestro sistema numérico, conservando los genes de sus padres tras multiplicarse. Así, por ejemplo, el número 2.187 es un número vampiro, al tener los mismos genes (dígitos) que sus dos progenitores, 27 y 81 ($27 \cdot 81 = 2.187$).

Los números vampiro *verdaderos* (con pedigrí) cumplen cuatro condiciones:

- Tienen un número par de dígitos.
- Se obtienen al multiplicar dos números, llamados *colmillos*, que tienen la mitad de dígitos que el original.
- Tienen los mismos dígitos que los colmillos, y en la misma cantidad (aunque en cualquier orden).
- Los colmillos *no* acaban simultáneamente en 0.

Se han detectado también algunos *números vampiro primos* por serlo sus colmillos (536.539 = 563 x 953), e incluso vampiros con varios pares de colmillos (125.460 = 204 x 615 = 246 x 510). Algunos números se limitan a imitar a los vampiros, como el 126 (6 x 21) cuyos colmillos no tienen la misma longitud, o el 1.395 (5 x 9 x 31) que tiene tres colmillos. Afortunadamente, los imitadores no son peligrosos, y no nos interesan.

Como con los vampiros humanos, los números vampiro no son fáciles de detectar. ¿Puedes ayudarnos?

Entrada

La entrada estará compuesta de un primer valor que indicará el número de casos de prueba. Cada caso de prueba contendrá un número a ser comprobado. Los números no tendrán más de 9 dígitos (en base 10).

Salida

Para cada caso de prueba el programa escribirá `SI` si el número es un vampiro *verdadero*, y `NO` si no lo es.

Entrada de ejemplo

```
4
2187
126
1122
536539
```

Salida de ejemplo

```
SI
NO
NO
SI
```

Fuente

Los números vampiro fueron sacados a la luz por primera vez por Clifford A. Pickover en el grupo `sci.math` de Usenet en 1994. El mensaje original puede ser visto en `http://groups.google.com/group/sci.math/msg/f17b2281a4aa16da?lr=&ie=UTF-8`

I

Divisores del factorial

El factorial de un número n es el resultado de multiplicar todos los números entre 1 y el propio n:

$$fact(n) = 1 * 2 * ... * (n-1) * n$$

Además, $fact(0) = 1$.

La pregunta que hoy nos hacemos *no* es cuál es el factorial de un número; ni siquiera cuál es el último dígito del factorial. La pregunta que nos hacemos es: si dividimos $n!$ por un número p, ¿el resto es cero?

Por ejemplo, el factorial de 6 es 720. Si lo dividimos por 8 el resto es cero pero si lo dividimos por 42 no.

Entrada

La entrada estará compuesta de varios casos de prueba. Cada caso de prueba consiste en una línea con dos números positivos, p y n, menores que 2^{31}, siendo p un número primo[4]. La entrada termina con un caso de prueba donde ambos números son negativos; para este último *no* se generará salida alguna.

Salida

Para cada caso de prueba, el programa escribirá `YES` si p divide a $n!$ y `NO` en caso contrario. Cada veredicto irá en una línea diferente.

Entrada de ejemplo

```
2 5
7 500000
7 3
-1 -1
```

Salida de ejemplo

```
YES
YES
NO
```

[4]En este problema, consideraremos el 1 como primo.

J

Una, dola, tela, catola...

Una tarde, diez peregrinos se detuvieron en una posada y solicitaron pasar allí la noche, pero el posadero sólo tenía alojamiento para cinco de ellos. Decidieron echar a suertes quién dormiría en cama y quién no, y para ello utilizaron la famosa cantinela:

"Una, dola, tela, catola,
quila, quilete,
estaba la reina en su gabinete,
vino Gil apagó el candil,
candil, candilón,
cuenta las veinte que las veinte son
policia y ladrón
uno, dos, tres..."

Los peregrinos se situaron colocados en círculo y uno de ellos comenzó a cantar; por cada palabra iba señalando a un peregrino según el orden de colocación. Al terminal la canción, el peregrino al que señalaba el dedo quedaba descartado. Ese peregrino elegido por la cancioncilla salía del círculo y sabía que tendría la desdicha de dormir en el suelo de la taberna, aunque, al menos, al lado del fuego. Para elegir al siguiente peregrino, el conteo comenzaba otra vez por la persona que seguía en el círculo a la recién eliminada.

Realiza un programa que, dado el nombre de varias personas, el número de camas disponibles y el número de palabras de la canción, proporcione el nombre de aquellas que dormirán en una cama.

Entrada

La entrada comenzará con un número que indicará cuántos casos de prueba que hay que procesar. A continuación, para cada uno de ellos recibiremos la lista con los nombres de los peregrinos (que acabará con el nombre ficticio `F`), un entero que nos dirá el número de camas disponibles y un entero mayor que cero que nos dirá el número de palabras de la canción.

Ten en cuenta que en una compañía de peregrinos nunca viajan más de 50 personas. Además, ninguno de los nombres de los peregrinos contiene espacios, ni supera las 25 letras.

Salida

Para cada caso de prueba escribiremos en una única línea el nombre o nombres de las personas que duermen en cama separados por espacios; el orden de los nombres será el mismo en el que están colocados en la entrada.

Si ninguno queda fuera, en vez de escribir todos los nombres se escribirá `TODOS TIENEN CAMA`. Si, al contrario, no hay camas en la posada para los peregrinos, se mostrará el mensaje `NADIE TIENE CAMA`.

Entrada de ejemplo

```
4
Anastasio Ignacio Felipe Borja Daniel Cesar F 2 3
Javier Ramiro Luis Rosa Carmen Paola Josefa F 0 3
Petra Santiago Pepi F 2 20
Merche Juanjo Miriam Pilar Marina Ovidio Rafael Eustaquio F 4 7
```

Salida de ejemplo

```
Anastasio Daniel
NADIE TIENE CAMA
Petra Pepi
Merche Miriam Pilar Marina
```

Fuente

Basado en *El Acertijo del Alguacil* del libro "Los Acertijos de Canterbury" (Henry E. Dudeney)