PISCINA

Se acerca el verano y llega el momento de sacar de los armarios y trasteros las piscinas para los niños (y no tan niños), colocarlas en la terraza, patio o jardín y llenarlas de agua para que los pequeños de la casa puedan empezar a disfrutarlas.
Este año la tarea se presenta complicada porque durante el invierno la larga manguera que permitía llevar el agua desde el grifo de la cocina hasta la propia piscina se ha perdido y habrá que hacerlo con un barreño…
Para complicar aún más las cosas, también durante el invierno la piscina (a pesar de ser de fibra de vidrio) se ha pinchado y pierde un poco de agua. Aun así, como los pequeños están ansiosos por darse un chapuzón decidimos llenarla cuanto antes, con pinchazo incluido, y luego mientras ellos disfrutan lo arreglaremos. Dado que la piscina está perdiendo agua constantemente, estará llena únicamente durante un instante de tiempo. En ese preciso momento dejaremos de hacer viajes a la cocina y nos pondremos rápidamente a arreglarla.
Como dice el refrán "mal de mucho consuelo de tontos"; la tarea de llenado será un poco más llevadera gracias al consuelo de saber que nuestro vecino está en la misma situación. A través del seto del jardín podemos verle haciendo viajes como un loco de su cocina a su piscina, para compensar el pinchazo que también él tiene. La pregunta es ¿quién tardará menos en llenar la piscina?
Cada uno de los casos de prueba empezará con la descripción de una "competición" entre nosotros y el vecino. Los tres primeros números indican los litros de agua de nuestra piscina ($1 \leq p \leq 109$), el número de litros de nuestro barreño ($1 \leq b \leq 109$) y por último los litros de agua que la piscina pierde durante el viaje. A continuación aparecen tres números para indicar la misma información pero de nuestro vecino.
Para cada caso de prueba aparecerá una línea compuesta por el ganador y el número de viajes que ha realizado. Como ganador, se indicará YO si nosotros terminamos antes de llenar la piscina (hacemos menos viajes para conseguirlo) y VECINO si es el vecino. En caso de empate, se indicará EMPATE.

Ejemplos
Entrada: 10 5 1 15 6 1
Salida: EMPATE 3

Entrada: 50 5 1 50 5 0
Salida: VECINO 10

Entrada: 50 5 1 50 5 6
Salida: YO 13

ASCENSOR

Las máquinas son incansables; una vez puestas en marcha, estarán funcionando sin quejarse por muchas veces que les pidas hacer lo mismo.

Pensemos en un ascensor, por ejemplo. Empieza el día en el bajo; el del primero le llama y le hace subir al quinto. Justo después le llaman en el segundo y le toca bajar a la planta baja. Luego el del tercero quiere ir a ver al del noveno... Al final del día ha hecho un largo recorrido de arriba a abajo, y de abajo a arriba. ¿Cuál es la longitud de ese recorrido?

Entrada
La entrada contiene distintos números enteros. El primero marca el piso inicial en el que empieza el ascensor (el 0 marca la planta baja; no hay plantas por debajo de ella). A continuación aparecen parejas de enteros, cada uno de ellos representando el uso del ascensor por parte de un vecino, con el piso desde el que llama al ascensor y con el piso destino. Se termina con un -1.

Salida
Para cada caso de prueba se mostrará una línea en la que aparecerá la longitud (en número de pisos) del recorrido completo del ascensor a lo largo del día.

Ejemplos:
Entrada: 0 1 5 2 0 3 9 -1
Salida: 19

5 5 4 -1
Salida: 1

1 2 3 4 5 -1
Salida: 4

CONSTANTE DE KAPREKAR

El matemático indio Dattaraya Ramchandra Kaprekar descubrió en 1949 una curiosa característica del número 6174. Hoy, se conoce a dicho número como constante de Kaprekar en honor a él.

El número es notable por la siguiente propiedad:

1. Elige un número de cuatro dígitos que tenga al menos dos diferentes (es válido colocar el dígito 0 al principio, por lo que el número 0009 es válido).
2. Coloca sus dígitos en orden ascendente y en orden descendente para formar dos nuevos números. Puedes añadir los dígitos 0 que necesites al principio.
3. Resta el menor al mayor.
4. Vuelve al paso 2.

A este proceso se le conoce como la rutina de Kaprekar, y siempre llegará al número 6174 en, como mucho, 7 iteraciones. Una vez en él, el proceso no avanzará, dado que 7641 − 1467 = 6174.

Por ejemplo, el número 3524 alcanzará la constante de Kaprekar en 3 iteraciones:

5432 − 2345 = 3087
8730 − 0378 = 8352
8532 − 2358 = 6174

Los únicos dígitos de cuatro cifras para los que la rutina de Kaprekar no alcanza el número 6174 son los repdigits, es decir aquellos cuyas cuatro cifras son iguales (como 1111), pues en la primera iteración se alcanzará el valor 0 y no podrá salirse de él. Es por esto que en el paso 1 se pedía explícitamente que el número inicial tuviera al menos dos dígitos diferentes.

El resto de los números de cuatro cifras terminarán siempre en el número 6174.

A continuación, se muestran dos ejemplos más:

- El número 1121 necesita 5 iteraciones:

  2111 − 1112 = 0999
  9990 − 0999 = 8991
  9981 − 1899 = 8082
  8820 − 0288 = 8532
  8532 − 2358 = 6174

- El número 1893 necesita 7:

  9831 − 1389 = 8442
  8442 − 2448 = 5994
  9954 − 4599 = 5355
  5553 − 3555 = 1998
  9981 − 1899 = 8082
  8820 − 0288 = 8532
  8532 − 2358 = 6174

Entrada
Un número a comprobar.

Salida
El programa indicará el número de vueltas que se debe dar a la rutina de Kaprekar para alcanzar el 6174.

ENCRIPTACIÓN DE MENSAJES

Uno de los métodos más antiguos para codificar mensajes es el conocido como cifrado Cesar. Su funcionamiento es simple: cada una de las letras del mensaje original es sustituida por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto.

Así, si utilizamos un desplazamiento de 2, las apariciones de la letra 'a' se sustituyen por la 'c', todas las apariciones de la 'b' por 'd', etc. El método tradicional comienza de nuevo al llegar al final del alfabeto, de forma que, con el desplazamiento de 2, la 'y' se sustituye por la 'a' y la 'z' se sustituye por la 'b'.

Los desplazamientos también pueden ser negativos; si utilizamos un desplazamiento de -1, la 'E' se convertirá en 'D', mientras que la 'a' pasará a ser 'z'.

Nuestro cifrado Cesar no codifica los caracteres que no sean letras anglosajonas. Así, por ejemplo, los espacios o los símbolos de puntuación no sufrirán cambio alguno.

Entrada
una única línea cuyo primer carácter es el código de la letra 'p', seguido de un mensaje codificado con el método Cesar descrito antes utilizando el desplazamiento adecuado para que la letra 'p' se codifique con ese primer carácter.

Salida
Mensaje descodificado

Ejemplo: Descodificar las siguientes líneas:

- qbfjpvBFJPV
- ozdhntZDHNT
- xXzwoziui-Um
- qGJO

CÓDIGOS DE BARRAS

En el lejano 1952, tres norteamericanos patentaron lo que terminó llamándose código de barras. Consiste en una técnica para representar números (y, en menos ocasiones, letras) mediante una serie de líneas verticales paralelas, con diferentes grosores y separaciones entre ellas. Si bien el primer uso sirvió para identificar de manera automática los vagones de un ferrocarril, hoy los códigos de barras se utilizan en infinidad de lugares, siendo la catalogación de productos la más habitual.

La manera concreta de codificar mediante barras los números y las letras puede ser muy variada, lo que ha llevado a la aparición de diferentes estándares. De todos ellos, el EAN (European Article Number) resulta ser el más extendido. De éste, hay principalmente dos formatos, que se diferencian en el ancho. Existe así el llamado EAN-8, que codifica 8 números, y el EAN-13, que, naturalmente, codifica 13.

6583 9522

(a) EAN-8

8 414533 043847

(b) EAN-13

Figura 1: Códigos de barras EAN

El último dígito del código se utiliza para detección de errores, y se calcula a partir de los demás. Para eso:
Empezando por la derecha (sin contar el dígito de control que se está calculando), se suman los dígitos individuales, multiplicados por un factor:

- Los dígitos en posiciones impares (empezando a contar por la derecha saltándonos el de control) se multiplican por 3.
- Los dígitos en posiciones pares se multiplican por 1.

Por ejemplo, para el código EAN-8 de la figura la operación a realizar es:
$2 \cdot 3 + 5 \cdot 1 + 9 \cdot 3 + 3 \cdot 1 + 8 \cdot 3 + 5 \cdot 1 + 6 \cdot 3 = 88$

El dígito de comprobación es el número que hay que sumar al resultado anterior para llegar a un valor múltiplo de 10. En el ejemplo de EAN-8, para llegar al múltiplo de 10 más cercano por encima del número 88 hay que sumar 2 (y llegar al 90). Ten en cuenta que si la suma resulta ser ya múltiplo de 10, el dígito de control será 0.

En EAN-13, los primeros dígitos se usan además para identificar al país. A continuación se indica una tabla (parcial) de esos códigos de país.

Código	País
0	EEUU
380	Bulgaria
50	Inglaterra

Código	País
539	Irlanda
560	Portugal
70	Noruega

Código	País
759	Venezuela
850	Cuba
890	India

Entrada
Cada caso de prueba contendrá un número perteneciente a un código de barras EAN-8 o EAN-13, incluyendo el dígito de control. Si el número de dígitos es inferior a 8, se asumirá que es un código EAN-8; si es superior a 8 pero inferior a 13, se asumirá EAN-13.

Salida
Para cada caso de prueba, el programa indicará si el dígito de control es correcto o no. Si lo es, escribirá "SI". En otro caso, escribirá "NO".

Si el código de barras es EAN-13 y correcto, el programa escribirá además el país al que pertenece utilizando la tabla anterior (separado por un espacio). Si el código no aparece en la tabla, el programa mostrará "Desconocido".

Ejemplos:
Entrada: 65839522
Salida: SI

65839521
Salida: NO

8414533043847
Salida: SI Desconocido

5029365779425
Salida: SI Inglaterra

5129365779425
Salida: NO

ESTROFAS

Historial ayer borrado,
anteayer hubo pecado.

El texto anterior es un pareado: una estrofa con dos versos que riman entre sí con rima consonante. ¿Sabrías hacer un programa que identifique distintos tipos de estrofa?
En concreto, nos bastará con identificar las rimas (no tendremos en cuenta el número de sílabas de cada verso), existiendo dos rimas distintas:

- Rima consonante: se dice que entre dos versos hay rima consonante cuando todos los sonidos, tanto vocales como consonantes, riman. Para las comparaciones se tienen en cuenta todos los sonidos a partir de la última vocal acentuada.
- Rima asonante: similar a la anterior pero únicamente riman las vocales.

Por ejemplo, el siguiente cuarteto de Diego de Silva y Mendoza:

Una, dos, tres estrellas, veinte, ciento, (A)
mil, un millón, millares de millares, (B)
¡válgame Dios, que tienen mis pesares (B)
su retrato en el alto firmamento! (A)

tiene esquema ABBA consonante, pues coinciden las vocales y consonantes del primer y último verso, así como las del segundo y tercero.
Nos piden ser capaces de identificar los siguientes tipos de estrofa:

- De dos versos:
  - Pareado: rima consonante AA.
- De tres versos:
  - Terceto: rima consonante en el primer y último verso (A-A). Ten en cuenta que AAA no se considerará terceto.
- De cuatro versos:
  - Cuarteto: rima consonante ABBA.
  - Cuarteta: rima consonante ABAB.
  - Seguidilla: rima asonante en los pares (-a-a). Ten en cuenta que otras combinaciones con más rimas o con rima consonante en lugar de asonante (por ejemplo -aaa o -A-A) no se consideran seguidillas.
  - Cuaderna via: rima consonante igual en todos los versos (AAAA).

Cada caso de prueba comienza con una línea que contiene un único entero con el número de versos del siguiente poema. A continuación aparecen tantas líneas como versos contiene la estrofa a analizar. Podemos asumir que la última palabra de cada verso es llana (la vocal acentuada está en la penúltima sílaba), y que ninguno tendrá más de 70 letras. La entrada no contendrá tildes para facilitar la programación, aunque esto signifique cometer errores ortográficos. Tampoco tendremos en cuenta que distintos elementos gráficos pueden tener el mismo sonido. Es decir, un verso terminado en -aba, no rimará de forma consonante con un verso terminado en -ava.
Para cada caso de prueba el programa indicará el nombre de la estrofa, utilizando mayúsculas (PAREADO, TERCETO, CUARTETO, CUARTETA, SEGUIDILLA, CUADERNA VIA) o la palabra DESCONOCIDO si no conoce la estrofa dada.

Ejemplos:

Número de Versos: 2
Poema:

Historial ayer borrado
anteayer hubo pecado

Resultado: PAREADO

Número de Versos: 2
Poema:

Esto no pega
ni con cola.

Resultado: DESCONOCIDO

Número de Versos: 4
Poema:

Era un simple clerigo, pobre de clerecia
dicie cutiano missa de la sancta Maria
non sabie decir otra, diciela cada dia
mas la sabie por uso qe por sabiduria

Resultado: CUADERNA VIA

Número de Versos: 3
Poema:

Un manotazo duro, un golpe helado
un hachazo invisible y homicida
un empujon brutal te ha derribado

Resultado: TERCETO

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 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).

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.

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 (5×5), otro de 9 (3×3) y otro de 1 (1×1):

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

Esta formación requería un total de 71 escudos.

Cada caso de prueba indicará el número de legionarios en el ejército que se quiere poner en formación.

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

Ejemplos:

Entrada: 35
Salida: 71

Entrada: 20
Salida: 44

Entrada: 10
Salida: 26

¡A DIBUJAR HEXÁGONOS!

Implementa un programa que, dado un entero n y un carácter c, dibuje el hexágono regular de n caracteres c de lado.
Cada caso de prueba es una línea con un entero mayor o igual que 0, seguido del carácter con el que se formará el hexágono.
Para cada caso de prueba se mostrará el hexágono regular que tiene como longitud de lado el número de caracteres especificado. Dicho hexágono estará dibujado con el carácter proporcionado en la entrada.

Ejemplos:

Entrada:
3 *
Salida:
```
  ***
*****
*****
   **
*****
  ***
```

Entrada:
4 p
Salida:
```
   pppp
pppppp
pppppppp
pppppppp
pp
pppppppp
  pppp
   pp
   ppp
   p
```