"First, solve the problem. Then, write the code" John Johnson.

ProgramaMe 2013
Final Nacional
Problemas

Patrocinado por

Universidad Complutense de Madrid

Ejercicios realizados por

Universidad Complutense de Madrid

I.E.S. Antonio de Nebrija (Móstoles)

Realizado en la **Facultad de Informática (U.C.M.)**
7 de junio de 2013

Listado de problemas

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Patricia Díaz García (I.E.S. Antonio de Nebrija - Móstoles)

Revisores:

- Ferran Borrell Micola (I.S. Baix Camp - Reus)
- Cristina Gómez Alonso (I.S. Baix Camp - Reus)
- Roger Meix Mañá (I.S. Baix Camp - Reus)

A
Juego con números

Zipi y Zape, cansados de hacer travesuras, se han inventado un nuevo juego y pasan horas jugando entre ellos. Visto desde fuera el juego tampoco es demasiado divertido, pero a ellos parece encantarles.

En cada partida, ambos eligen tres números positivos, por ejemplo:

```
Zipi: 4 7 12
Zape: 3 7 8
```

Después los ponen todos seguidos, `4712378` y empieza la partida. En cada turno, uno de ellos suma 1 a cada número de forma que el 4 se convierte en 5, el 7 en 8, el 12 en 13, etc. Al ponerlos de nuevo todos juntos queda: `5813489`. En el siguiente turno le toca al siguiente, que vuelve a sumar uno a cada número, después al siguiente, y así hasta llega a sumar 10. Cuando, al colocar todos los números seguidos, la cadena *incrementa su longitud*, el jugador que ha sumado gana un punto. Tras los 10 turnos, gana el que tiene más puntos.

La partida completa del ejemplo es:

```
Inicio: 4712378
Zipi  : 5813489
Zape  : 69145910 => Al sumar 2, Zape gana un punto
Zipi  : 7101561011 => Al sumar 3, Zipi gana un punto
Zape  : 8111671112
Zipi  : 9121781213
Zape  : 10131891314 => Al sumar 6, Zape gana un punto
Zipi  : 111419101415 => Al sumar 7, Zipi gana un punto
Zape  : 121520111516
Zipi  : 131621121617
Zape  : 141722131718
```

En este caso el resultado final es empate a dos.

Ahora quieren ampliar las reglas de forma que en vez de tres números cada uno elija cuatro. Y en vez de que haya diez turnos (sumar hasta diez), que sumen hasta diez millones (10.000.000). Para evitar que la partida sea demasiado larga, preferirían tener un programa que les escribiera el desarrollo de la misma una vez elegidos los números...

Entrada

La entrada estará compuesta por distintos casos de prueba que representan distintas partidas. Cada uno de ellos consiste en dos líneas con 4 números cada una; la primera contiene los números elegidos por Zipi y la segunda los elegidos por Zape, todos ellos mayores que cero.

La entrada termina con una línea con un único número, el cero, que no debe ser procesada.

Salida

Para cada caso de prueba se mostrará la evolución de la partida y el resultado final según el formato del ejemplo. En concreto, aparecerá una línea cada vez que uno de los hermanos gana un punto que contendrá el valor sumado a todos los números y quién se lleva el punto, de forma que el valor sumado deberá aparecer en orden creciente. Tras eso aparecerá el marcador final de la partida. Revisa el ejemplo de salida para más detalles.

Entrada de ejemplo

```
10000000 10000000 10000000 10000000
9999999 10000000 10000000 10000000
100 123456 345678 9999993
200 999994 999995 999996
0
```

Salida de ejemplo

```
Zipi: 1
Resultado final: Zipi: 1 Zape: 0
Zape: 4
Zipi: 5
Zape: 6
Zipi: 7
Zape: 800
Zape: 900
Zape: 9800
Zape: 9900
Zape: 99800
Zape: 99900
Zape: 654322
Zape: 876544
Zape: 999800
Zape: 999900
Zape: 9000004
Zipi: 9000005
Zape: 9000006
Zape: 9654322
Zape: 9876544
Zape: 9999800
Zape: 9999900
Resultado final: Zipi: 3 Zape: 18
```

B
Quinto milenio

Túker Chiménez ve cosas escondidas en cualquier lado. Una mancha de humedad en la pared se le antoja la cara de la anterior propietaria de una casa; el sonido del viento le parece un susurro venido del más allá; una nube con forma peculiar le convence de la existencia de vida extraterrestre.

Ahora le ha dado por ver mensajes ocultos en cualquier sitio. Por poner un ejemplo, si lee el siguiente titular: "*El presidente del Gobierno se somete esta noche al escrutinio de varios periodistas en Televisión Española.*", se las ingenia para leer un "te odio" oculto que le mantiene en vela toda la noche:

*El presiden***T***e d***E***l Gobierno se s***O***mete esta noche al escrutinio* ***D****e var***I***os peri***O***distas en Televisión Española.*

Ahora quiere automatizar la tarea de la búsqueda de estos mensajes.

Entrada

La entrada comenzará con un entero que indica el número de casos de prueba. Cada uno de ellos está formado por dos líneas; la primera indica el *titular* donde buscar un mensaje oculto y la segunda indica el mensaje a buscar. La longitud de cada una de las cadenas no excederá los 2000 caracteres. Ten en cuenta que no hace falta distinguir entre mayúsculas y minúsculas y que los espacios del mensaje oculto no son relevantes, es decir, no hace falta que existan en el mensaje original, pero sí deben aparecer el resto de caracteres (signos de puntuación, comillas, etc.). Se garantiza que el mensaje oculto a buscar no empieza ni termina por espacios.

La entrada contendrá únicamente letras del alfabeto inglés, por lo que no aparecerán vocales con tilde. Además, podrían aparecer múltiples espacios consecutivos.

Salida

Para cada caso de prueba el programa escribirá `SI` si el mensaje buscado aparece en el titular y `NO` en caso contrario.

Entrada de ejemplo

```
4
...dente ...somete ... de varios periodistas ...
te odio.
Teo dijo "si".
te odio.
Y adios, que ya viene el alba.
te odio.
Teo    subio  al    podio.
te          odio.
```

Salida de ejemplo

```
SI
SI
NO
SI
```

C
Molinos de viento

Los molinos de viento aprovechan la fuerza del viento para conseguir energía. Las empresas eléctricas los instalan en parajes donde la cantidad de este fenómeno natural es grande durante todo el año, para garantizar que cada uno de ellos sea económicamente rentable.

El mantenimiento de estas altas construcciones, lamentablemente, no resulta barato. Tanto es así que una empresa ha decidido desprenderse de unos cuantos de estos molinos, vendiéndolos a otra empresa del sector para que sea ella la que cubra con todos los gastos y, si tiene suerte, con los beneficios.

Esa nueva empresa ha establecido que lo más rentable es comprar, de toda la hilera de molinos que están a la venta, un puñado de ellos *consecutivos*, porque así se minimizan los desplazamientos que el personal de mantenimiento tiene que hacer entre los molinos que supervisa.

Por lo tanto, ahora le toca elegir qué molinos quiere comprar. Para eso tiene los datos de rendimiento de cada uno de ellos (la cantidad de energía producida) y quiere saber *rápidamente* cuánta energía produciría un intervalo concreto de molinos.

Entrada

La entrada estará compuesta de varios casos de prueba. Cada caso de prueba contiene la descripción de un campo de molinos y una serie de intervalos de los mismos de los que se quiere conocer la cantidad total de energía producida.

En concreto, cada caso de prueba comienza con un número indicando la cantidad de molinos instalados (hasta 1.000.000). A continuación se indica, en otra línea, la cantidad de energía producida por cada uno de ellos que será siempre mayor o igual que `0`. El orden de los molinos es el mismo que aparece en el campo de molinos, es decir, los molinos más cercanos geográficamente también aparecen juntos en la entrada.

Tras la descripción de la energía producida vendrá un número que indica el número de consultas que la empresa compradora hará. A continuación, en líneas independientes, aparecerán dos números que indican el número de molino inicial y molino final que marcan el intervalo del que se quiere conocer la energía total producida. El número del molino inicial *siempre* sera menor o igual que el del molino final. La entrada termina con un caso de prueba sin molinos que no deberá ser procesado.

Salida

Para cada caso de prueba se debe escribir una línea por cada consulta realizada por la empresa compradora en la que se indicará la energía total producida por los molinos en ese intervalo.

Entrada de ejemplo

```
5
5 6 7 7 1
3
1 5
1 3
5 5
0
```

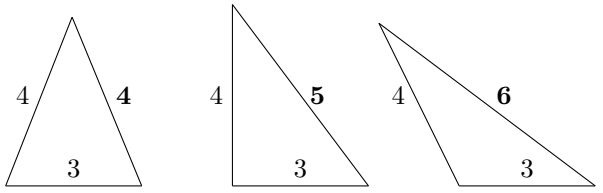
Salida de ejemplo

```
26
18
1
```

D
Triángulos

Es bien sabido que la suma de los ángulos de cualquier triángulo es siempre 180 grados. En función del ángulo *mayor*, los triángulos se pueden clasificar en tres tipos:

- *Acutángulo*: el mayor de los tres ángulos es *agudo* (menor de 90 grados).
- *Rectángulo*: el mayor de los tres ángulos es *recto* (exactamente 90 grados).
- *Obtusángulo*: el mayor de los tres ángulos es *obtuso* (mayor de 90 grados).



Acutángulo Rectángulo Obtusángulo

¿Eres capaz de, a partir de la longitud de tres segmentos, decir el tipo de triángulo que forman?

Entrada

La entrada consistirá en un primer número indicando el número de casos de prueba que vendrán después.

Cada caso de prueba ocupará una línea, y estará compuesto de tres números enteros no negativos menores que $2^{15} - 1$, separados por espacios y no necesariamente ordenados. Cada entero representará la longitud de cada uno de los lados de un triángulo.

Salida

Para cada caso de prueba, el programa indicará el tipo de triángulo, escribiendo `ACUTANGULO`, `RECTANGULO` u `OBTUSANGULO`. Si resulta imposible formar un triángulo con esos segmentos, escribirá `IMPOSIBLE`.

Entrada de ejemplo

```
4
3 4 4
5 3 4
3 4 6
3 4 7
```

Salida de ejemplo

```
ACUTANGULO
RECTANGULO
OBTUSANGULO
IMPOSIBLE
```

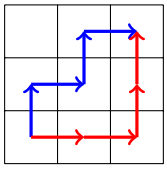
Fuente

Inspirado en un comentario de *La proporción áurea. El lenguaje matemático de la belleza*, de Fernando Corbalán.

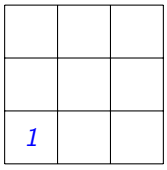
E
Las torres perezosas

En un tablero de ajedrez las torres pueden moverse en horizontal y en vertical tantas casillas como quieran, pero siempre en línea recta. En nuestro caso, tenemos una torre perezosa que también se mueve en horizontal y vertical, pero únicamente una casilla cada vez.

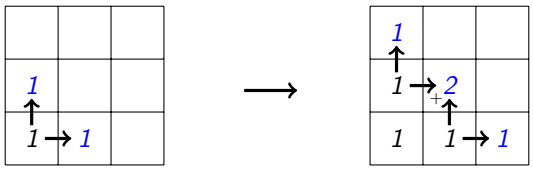
Si situamos a nuestra torre en la esquina inferior izquierda de un tablero de 3x3 y queremos que vaya a la esquina superior derecha, es fácil ver que tendrá que hacer como mínimo 4 movimientos, aunque puede seguir varias rutas distintas. En la siguiente figura aparecen dos de ellas:



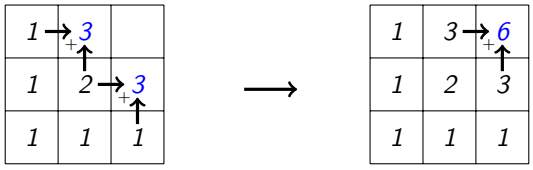
Tiene sentido plantearse, ¿de cuántas formas distintas podría llegar nuestra torre desde el origen al destino con el mínimo número de movimientos posible? Una forma de hacer el cálculo es averiguar de forma iterativa el número de formas diferentes que tenemos de llegar hasta cada uno de los escaques intermedios. Empezamos anotando que sólo hay una forma para que la torre "vaya" de la casilla inicial a esa misma casilla inicial en el mínimo número de movimientos (no moviéndose):



Para las demás, sabemos que el número de maneras de llegar a un escaque en el mínimo número de pasos será la suma del número de formas de llegar a cada uno de los escaques adyacentes hacia la izquierda o hacia abajo (llegar a un escaque por arriba o por la derecha sería "ir hacia atrás"). Por tanto, lo que tenemos que hacer es ir calculando de cuántas formas se puede llegar a los escaques adyacentes de los ya conocidos sumando los números conseguidos anteriormente:

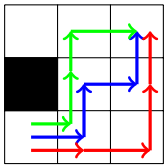


Repitiendo el proceso hasta llegar a la última celda tenemos:



Esto demuestra que en un tablero de 3x3 hay seis formas distintas de ir, con el mínimo número de movimientos posibles, de una esquina a otra.

Si en el tablero hay algunas casillas *no transitables* es necesario adaptar el recorrido para tenerlas en cuenta. Por ejemplo, en el siguiente tablero de 3x3 con una celda no transitable se puede llegar del origen al destino de tres formas distintas utilizando los cuatro movimientos mínimos:



Entrada

La entrada estará compuesta de varios tableros de distintos tamaños. Cada tablero comienza con una línea que contiene un número n ($1 \leq n \leq 15$) que indica el número de filas y columnas del tablero. A continuación aparecen n líneas, una por cada fila con n caracteres cada una. El carácter `.` (punto) indica una casilla transitable, mientras que `X` significa casilla no transitable. Se garantiza que las casillas origen y destino siempre serán transitables.

La entrada termina con un tablero vacío que no debe procesarse.

Salida

Para cada caso de prueba se escribirá una línea con un único número que indica de cuántas formas distintas puede llegar nuestra torre perezosa desde la esquina inferior izquierda del tablero a la esquina superior derecha pasando por el mínimo número de casillas.

Se garantiza que el resultado nunca será mayor que 10^9.

Entrada de ejemplo

```
3
...
...
...
3
...
X..
...
0
```

Salida de ejemplo

```
6
3
```

F
Persistencia multiplicativa de los números

En 1973, el matemático inglés Neil Sloane definió, en una revista dedicada a las matemáticas recreativas, la *persistencia multiplicativa* de los números. Consiste en el número de veces que hay que multiplicar los dígitos de un número (escrito en base 10) hasta llegar a un número de un único dígito.

Por ejemplo, el número 39 tiene una persistencia multiplicativa de 3:

$$39 \xrightarrow{3\cdot 9} 27 \xrightarrow{2\cdot 7} 14 \xrightarrow{1\cdot 4} 4$$

¿Eres capaz de calcular la persistencia multiplicativa para otros números?

Entrada

La entrada comenzará con un número indicando la cantidad de casos de prueba que vienen a continuación. Cada uno consistirá en una única línea con un número no negativo menor que 2^{31}.

Salida

Para cada caso de prueba se debe indicar, en una línea independiente, la persistencia multiplicativa del número.

Entrada de ejemplo

```
3
39
93
391
```

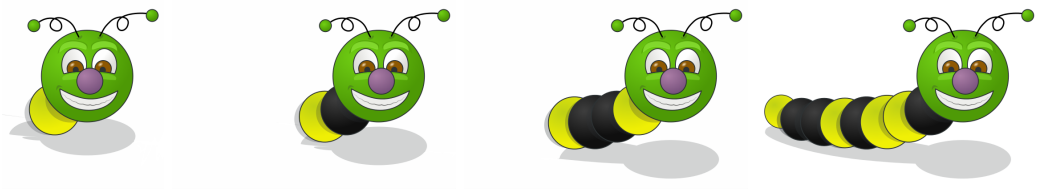
Salida de ejemplo

```
3
3
3
```

G
Anélidos

Los anélidos son una *variedad* de lo que coloquialmente se conoce como *gusanos*. Su propiedad más característica (y que los diferencia del resto de variedades) es que sus cuerpos están compuestos por la repetición de "anillos" con la misma estructura, repitiendose en cada anillo todos los órganos internos y externos. Seguramente los anélidos más conocidos sean las orugas.

En algunos lugares de la selva tropical se da una variedad de oruga, llamada *thuelis*, que intriga desde hace muchos años a los biólogos. Cuando nacen, poseen únicamente un anillo (además de la cabeza), de color o bien amarillo, o bien negro. En cada fase de crecimiento todos sus anillos se replican, de forma que donde antes había un anillo ahora aparecen dos. En particular cada anillo da lugar, a su lado y más cerca que él de la cabeza, a un nuevo anillo del color contrario:



Estas orugas crecen indefinidamente siguiendo ese patrón. Aunque no son venenosas, los colores que muestran son similares a los de otras especies que sí lo son, por lo que los depredadores suelen evitarlas. Se han llegado a encontrar ejemplares de casi dos metros.

Para una distribución inicial de anillos, ¿eres capaz de decir qué colores tendrán después de varias fases de crecimiento?

Entrada

La entrada estará compuesta de varios casos de prueba. Cada uno empezará con un número entero no negativo que indicará cuantas fases de crecimiento hay que simular. Después vendrá una cadena indicando los colores de los anillos del ejemplar (`A` para amarillo y `N` para negro), acabando con la letra `C` que indica la posición de la cabeza.

Ten en cuenta que, en muy raras ocasiones, el crecimiento no sigue el patrón habitual, por lo que las configuraciones iniciales de la entrada *no* necesariamente serán alcanzables desde alguno de las dos variedades de nacimiento de los *thuelis* (un único anillo amarillo o negro).

La entrada terminará cuando se reciba un *thuelis* sin anillos.

Salida

Para cada caso de prueba el programa deberá indicar, en una línea, la configuración de colores de los anillos tras el número solicitado de fases de crecimiento, asumiendo que todas siguen el patrón habitual descrito. Se usará, como en la entrada, `A` para amarillo y `N` para negro, acabando con la `C` para la cabeza. No se añadirán espacios en ningún caso.

Entrada de ejemplo

```
1 AC
1 NAC
2 NC
0 C
```

Salida de ejemplo

```
ANC
NAANC
NAANC
```

H
El sueño de los concursantes

Luis participa hoy en un concurso de programación. No llega en su mejor momento; tiene un sueño atroz. Ha estado los últimos días practicando ejercicios para maximizar sus posibilidades de triunfo pero quizá al haber dormido tan poco lo que haya conseguido haya sido lo contrario.

La esperanza que le queda es que es posible que el resto de participantes estén igual. ¿Le ayudas a calcular cuánto ha dormido cada concursante en las últimas noches?

Entrada

La entrada estará compuesta de múltiples casos de prueba. Cada caso de prueba consiste en las horas de sueño de un participante. Para cada uno de ellos aparece una primera línea que indica cuántas noches hay que procesar. A continuación aparece una línea para cada una de esas noches con la hora a la que se fue a dormir y la hora en la que se despertó, con el formato `HH:MM-HH:MM`. Ten en cuenta que ningún participante se fue a dormir nunca antes de las 22:00 ni se despertó después de las 10:00. Además, todas las noches durmieron al menos un minuto.

La entrada terminará cuando el número de noches a calcular del siguiente participante sea cero.

Salida

Para cada caso de prueba se expresará cuánto tiempo durmió el participante en horas y minutos. Tanto en las horas como en los minutos, si el valor es menor que `10` se pondrá un `0` delante para que tengan dos dígitos. Ten presente que el número de minutos deberá ser menor que 60.

Entrada de ejemplo

```
1
00:00-07:00
2
22:00-10:00
23:58-00:04
0
```

Salida de ejemplo

```
07:00
12:06
```

Potitos

"¡Javi! ¡Parece que éste tampoco le gusta!" se convirtió en una cantinela habitual a la hora de la comida, antes de que el pequeño dejara definitivamente de comer potitos una tarde. Darle de comer era misión imposible; pese a la paciencia de sus padres, el niño había heredado el carácter cabezota de su padre y mantenía la boca cerrada, inmune a los vuelos de la cucharita y a las monerías de aquellos que le rodeaban. Era necesario encontrar una solución.

Se les ocurrió que, quizá, el aparentemente caprichoso rechazo a los potitos se debía a algún ingrediente concreto. Comenzaron a anotar, cuidadosamente, todos los ingredientes de cada potito que le daban, junto con un comentario de si el pequeño se lo había tomado o no. Tras varios días, estaban convencidos de que podrían averiguar cuáles eran los ingredientes que no le gustaban, y así comprar aquellos potitos que fuera a comerse. ¿Puedes ayudarles a encontrarlos?

Entrada

La entrada estará compuesta de múltiples casos de prueba. Cada uno comienza con un número indicando el número de potitos que se han intentado dar al bebé (como máximo 25).

A continuación aparece una línea por cada potito. La línea comienza por `SI` o `NO` dependiendo de si el pequeño se comió o no el potito. Después aparece la lista de los ingredientes del potito separados por espacios. La lista se cierra con la palabra `FIN`, que no debe considerarse un ingrediente. Ningún potito tiene más de 10 ingredientes, y todos los ingredientes están compuestos por una única palabra de hasta 20 letras minúsculas.

La entrada acaba con un caso de prueba sin potitos.

Salida

Para cada caso de prueba se deben mostrar, en una línea, los ingredientes que no le gustan al niño, ordenados alfabéticamente y separados por espacio. Si todos los ingredientes le gustan, se dejará la linea en blanco.

Entrada de ejemplo

```
3
SI: patata maiz tomate FIN
NO: patata puerro guisantes pollo FIN
SI: tomate zanahoria puerro pollo calabacin arroz FIN
2
SI: tomate zanahoria pollo calabacin arroz FIN
NO: tomate ternera puerro FIN
0
```

Salida de ejemplo

```
guisantes
puerro ternera
```

J
Y el ganador es...

En muchos de los concursos de programación, como en el que hoy participas, cada vez que un equipo resuelve correctamente un problema recibe un globo del color asociado a ese problema. Al final, quien más globos consigue no sólo tiene su ordenador más colorido, sino que será el ganador del concurso.

Dada la lista de los globos colocados a cada equipo, ¿eres capaz de decir quién es el ganador?

Entrada

La entrada estará compuesta de múltiples casos de prueba, cada uno de ellos simulando un concurso. Cada caso de prueba comienza con una línea con dos números, el primero de ellos indicando el número de equipos participantes (hasta un máximo de 20) y el segundo el número de globos entregados.

A continuación aparecerá una línea por cada globo entregado, con el número del equipo que lo ha recibido (entre 1 y el número de equipos) y el color (una palabra de un máximo de 20 letras). Un equipo nunca recibirá dos veces el mismo color de globo.

La entrada terminará cuando se llegue a un concurso sin equipos.

Salida

Para cada caso de prueba se debe escribir el número del equipo ganador en una linea. En caso de empate, se escribirá `EMPATE`.

Entrada de ejemplo

```
4 3
2 Rojo
3 Amarillo
3 Azul
4 4
2 Rojo
3 Amarillo
3 Azul
2 Verde
0 0
```

Salida de ejemplo

```
3
EMPATE
```