



Concurso Regional Programame
Zaragoza
21 de Marzo de 2024

Cuaderno de Problemas



Índice de problemas

A. SIEMPRE CON PRISA	3
B. ORDENANDO LOS APUNTES	4
C. HACIENDO EQUIPO	5
D. PREPARANDO LA COMIDA	6
E. QUIERO SER UNA ESTRELLA DEL ROCK	7
F. ORGANIZANDO VUELTAS CICLISTAS	8
G. CUÁNTOS AÑOS TIENES	9
H. DENTRO DEL LABERINTO	10
I. EL AMIGO ALEMÁN	12
J. FELIZ 2024	14
K. NÚMEROS HARSHAD	15
L. QUEDANDO CON LA CUADRILLA	16

Ejercicios realizados por:

- Antonio Pérez-Aradros
- Santiago Faci

A. SIEMPRE CON PRISA

Diana es una de esas personas que siempre anda con prisa. Por ejemplo, siempre calcula qué ruta le será más rápida cuando está en un edificio: subir andando por las escaleras o coger el ascensor.



Así, siempre que Diana va a un sitio que no conoce, se preocupa de conocer todos los detalles para saber qué hacer una vez que esté ahí, subir por las escaleras o bien coger el ascensor. Por supuesto sube las escaleras de 2 en 2 siempre que puede. Además, se conoce bien y sabe que le cuesta un segundo dar cada zancada (ya sea para subir 2 escalones o solamente 1). Y también se documenta bien: sabe que la velocidad media de un ascensor es de 1 m/s, por lo que es la que usa como referencia para sus cálculos.

Como Diana es una chica de ciencia le gusta ser exacta y añade 1 segundo extra por piso cuando sube andando, que es el tiempo que le cuesta alcanzar, una vez que está en un rellano, el siguiente grupo de escaleras. Considera que está arriba cuando consigue subir el último peldaño del último piso.

Entrada

La entrada comienza con un número que indica el número de casos que vendrán a continuación.

Para cada caso se presenta la información que Diana necesita conocer de un edificio para hacer sus cálculos: El número de pisos, la altura de cada uno en metros y el número de peldaños que hay entre 2 pisos.

Salida

La salida contendrá, para cada caso de entrada, qué debería hacer Diana si tuviera que subir al último piso de dicho edificio. Se indicará **ANDANDO** en caso de que tenga que subir andando para llegar antes o **ASCENSOR** si la opción más rápida es coger el ascensor.

Entrada de ejemplo

```
4
5 3 10
6 4 12
8 3 15
10 5 10
```

Salida de ejemplo

```
ASCENSOR
ANDANDO
ASCENSOR
ANDANDO
```

B. ORDENANDO LOS APUNTES

Mala época la de los exámenes cuando eres estudiante. Diana sabe muy bien lo que es eso porque está estudiando 2ª Bachillerato y el miércoles tiene examen de Historia.

Ella es muy aplicada y ya ha empezado a estudiar. Pero justo cuando empezaba, se ha dado cuenta de que tiene los apuntes totalmente desordenados. Sabe cuál es el primer tema del examen y ha encontrado, entre sus papeles, anotaciones desordenadas de qué tema va después del tema justo anterior.



Así que su primera labor antes de ponerse a estudiar va a ser saber en qué orden debería estudiarse todo. Le gusta la programación y se estaba ahora mismo preguntando si ya existirá un programa que permita introducir el primer tema y todos esos fragmentos desordenados de parejas de temas, para que le muestre el orden de todos ellos. Sería muy práctico ahora mismo. ¿Valdrá la pena hacerlo en lugar de ordenar a mano todo ese jaleo?

Entrada

Como entrada se recibe el primer tema en la primera línea. A continuación se recibe el número de fragmentos desordenados de parejas de temas. Después de eso, se reciben las parejas de temas que sabe que van seguidos. Una de esas parejas de temas siempre tendrá, como primer tema, el que se ha indicado en la primera línea de la entrada.

Salida

La salida mostrará el orden correcto de todos los temas en una sola línea

Entrada de ejemplo

```
LA EDAD MEDIA
4
GUERRA COLONIAL-LA SEGUNDA REPUBLICA
EL SEXENIO REVOLUCIONARIO-GUERRA COLONIAL
REVOLUCION LIBERAL-EL SEXENIO REVOLUCIONARIO
LA EDAD MEDIA-REVOLUCION LIBERAL
```

Salida de ejemplo

```
LA EDAD MEDIA, REVOLUCION LIBERAL, EL SEXENIO REVOLUCIONARIO, GUERRA
COLONIAL, LA SEGUNDA REPUBLICA
```

* La salida de ejemplo es una sola línea

C. HACIENDO EQUIPO

Últimamente, debido al auge del trabajo en remoto, se han puesto de moda los retiros para hacer equipo. Son jornadas en las que se juntan compañeros de trabajo, que no se ven habitualmente, para hacer todo tipo de actividades que no tengan nada que ver con trabajar.



En San Valero no somos menos y estamos tratando de organizar una jornada para hacer equipo y pasar un rato juntos. Los propios profesores han propuesto una serie de actividades de las que se realizarán 2. Una vez que se tiene la lista preparada con todas las opciones, cada profesor ha de apuntarse a 2 de esas actividades.

Los criterios para escoger una actividad son bastante previsibles. Las 2 actividades más votadas serán las elegidas (los profesores que no las votaron tendrán que hacerlas igualmente, claro). Y en caso de empate, ganará la actividad que fue propuesta antes, que será la que aparece antes en el listado con respecto a la que empate.

Por supuesto, hasta que todo el mundo no haya votado no será posible emitir el veredicto. El sistema no generará los resultados hasta que todos los profesores hayan votado sus 2 actividades. Lo que seguro que no ocurre es que algún profesor haya votado solamente por una de las actividades, por lo que no vamos a contemplar esa situación.

Entrada

En primer lugar se recibe como entrada una línea con los nombres de todos los profesores que participan en las actividades. Y a continuación una serie de actividades (una por línea) junto con los nombres de los profesores que se han apuntado a la misma.

Salida

Como salida se devolverán las 2 actividades seleccionadas (una por línea). En caso de que todavía no haya votado todo el mundo, se devolverá la línea AUN NO HAN TERMINADO LAS VOTACIONES

Entrada de ejemplo

```
Alejandro Vanessa Diana Antonio Alex Alberto Borja Eduardo
Ir en bici - Alejandro Vanessa
Tiro con arco - Antonio Diana Eduardo Alex Alberto
Esquiar - Antonio Eduardo Borja Alejandro
Montar a caballo - Diana Alex Alberto Borja Vanessa
```

Salida de ejemplo

```
Tiro con arco
Montar a caballo
```

D. PREPARANDO LA COMIDA

En casa no nos gusta cocinar solos. Nos los pasamos mejor cocinando juntos y además ahorramos algo de tiempo porque podemos hacer hasta un par de cosas al mismo tiempo. La realidad es que tampoco tenemos una cocina muy grande, por lo que solamente podemos cocinar en parejas.

Lo que hacemos es preparar una lista de las tareas en el orden en que deberían llevarse a cabo y las repartimos. El primero que acaba lo que esté haciendo coge la siguiente tarea de la lista y así hasta que todo queda hecho. Si los 2 quedamos libres en el mismo momento, no pasa nada, puesto que no importa quién hace qué, solamente que se haga.



Y como nos gusta mucho tenerlo todo controlado, hemos pensado en hacer una aplicación a la que le pasemos toda esta información para que nos indique la hora a la que estará la comida. Así podemos avisar al resto de la casa para que estén sentados en la mesa en el momento exacto en que la comida estará lista.

Entrada

Como entrada se recibe, en primer lugar, la hora a la que se empieza a preparar la comida. A continuación vendrá un número que indica el número de pasos que hay que llevar a cabo. A partir de ahí se recibe, en cada línea, un texto que describe el paso a realizar y la duración en minutos. Podemos asegurar que todas las recetas contienen un número de pasos ≥ 2 .

Salida

La salida devolverá la hora a la que se terminará de cocinar. Tampoco hacemos unas recetas muy complicadas, por lo que la hora final nunca se nos junta con la merienda, que es a las 18:30.

Entrada de ejemplo

```
15:30
5
cocer la pasta 15
preparar la ensalada 10
freir la carne 10
poner la mesa 5
servir 3
```

Salida de ejemplo

```
15:53
```

E. QUIERO SER UNA ESTRELLA DE ROCK

Como Diana ya sabe tocar varios instrumentos, ha empezado a asistir a clases de canto para poder componer ella sola sus propias canciones.

Hace poco que ha empezado, por lo que por ahora solamente es capaz de tararear ritmos que ella misma inventa en sus ratos libres. Siempre que se le ocurre alguno lo escribe en su libreta, y ahora mismo ya tiene cientos de ellos. Como a estas alturas todavía no se le ocurre ninguna letra ni temática, todas se basan en diferentes secuencias de los 5 sonidos que le han enseñado en sus primeras clases: TARARA, TERERE, TIRIRI, TORORO y TURURU.



Y ahora, para ir mejorando, su profesor de canto le ha dado algunas directrices para asegurar que la canción esté bien compuesta y tenga buen ritmo. Una de esas directrices es que no puede repetir el mismo sonido más de 2 veces seguidas porque hace la canción un tanto repetitiva. También le ha pedido que pase a limpio todos esos ritmos porque se encuentran llenos de tachones y de otras palabras sin sentido que Diana ha ido probando pero que, por lo visto, no sirven.

Ahora ella se encuentra con cientos de ritmos que ha tarareado sin saber si cumplirán o no las reglas que su profesor le ha enseñado. Lo que ha pensado, para no descartar ninguna de ellas, es hacer los ajustes necesarios en todas ellas para que cumplan con esas reglas. Luego, cuando tenga más tiempo y práctica, ya buscará una letra adecuada, les añadirá música con los instrumentos y verá si suenan o no como ella espera.

Entrada

Como entrada se recibe una canción tal y como Diana la tararea cuando tiene sus momentos de inspiración.

Salida

Como salida se debe devolver la canción corregida para que cumpla con las directrices del profesor.

Entrada de ejemplo

```
TARARA TARARA TARARA LALALA TARARA TARARA TARARA TARARA TATATA TARARA  
TIRIRI TARARA TIRIRI TIRIRI TIRIRI TARARA TUTUTU TARARA LKJASLKJAD  
LKJASLKJAD LKJASLKJAD TIRIRI TARARA TURURU TURURU TURURU
```

Salida de ejemplo

```
TARARA TARARA TIRIRI TARARA TIRIRI TIRIRI TARARA TARARA TIRIRI TARARA  
TURURU TURURU
```

*Tanto entrada como salida son una sola línea

F. ORGANIZANDO VUELTAS CICLISTAS

La Federación Española de Ciclismo está probando cómo incluir la IA en la toma de decisiones, concretamente para decidir el itinerario definitivo en las vueltas cortas (de máximo 10 días de duración). La idea es ahorrarse los quebraderos de cabeza que da ajustar la combinación de etapas, kilómetros, días de competición y otros parámetros a tener en cuenta.



Así, lo que hará la organización será preparar el listado de posibles etapas para una vuelta determinada, pasarlo por esa IA que han creado y será esta aplicación quien acepte o descarte las etapas en función de una serie de parámetros que se le han enseñado. Estos parámetros son:

- Siempre habrá una etapa PROLOGO y otra FINAL de 50 kilómetros cada una. Como son etapas urbanas muy bien definidas, la IA no las tendrá en cuenta para comprobar si son o no correctas, pero sí para el cómputo total del número de etapas permitidas
- La vuelta no podrá tener nunca más de 10 etapas en total
- En una misma vuelta no puede haber más de 300 kms de un mismo tipo de etapa. Se rechazarán las etapas que supongan pasarse de esa cifra
- Una etapa nunca podrá tener más de 200 kms de longitud

Entrada

Como entrada se recibe la lista de etapas con el tipo y duración de cada una de ellas

Salida

Como salida se devolverá, para cada etapa, si ésta se acepta (OK) o se rechaza (KO) en función de los parámetros definidos. Al final se devolverá también el número total de kilómetros y el número de etapas teniendo en cuenta solamente las que la IA ha aceptado más las PROLOGO y FINAL.

Entrada de ejemplo

```
PROLOGO 50
PLANA 220
MONTANA 120
MEDIA-ALTURA 140
PLANA 120
ALTURA 120
PLANA 200
ALTURA 80
FIN 50
```

Salida de ejemplo

```
KO
OK
OK
OK
OK
KO
OK
680
7
```

G. CUANTOS AÑOS TIENES

- ¿En qué año naciste?
- En el 2002
- Entonces tienes 🤔 Ah, sí, 2024 – 2002: 22 años
- Pues no, todavía no los he cumplido porque estamos en marzo y mi cumpleaños es en diciembre. Así que tengo 21
- Oye, pues sabes lo que te digo, que entonces no es suficiente con saber el año de nacimiento y el año en el que estamos para saber los años que uno tiene
- Pues claro. A ver, ¿tú en qué año has nacido?
- Yo, en el 2003
- Entonces, seguro que tienes 20 o 21
- Sí, es verdad, acabo de cumplir 21



Entrada

El programa deberá leer, de la entrada estándar, múltiples casos de prueba, cada uno ocupando una única línea.

En cada línea aparecen dos números N, M ($2000 \leq N < M \leq 2100$) indicando el año de nacimiento de mi amigo y el año en el que nos encontramos.

La última línea son dos ceros que no deben procesarse.

Salida

Para cada caso de prueba, el programa escribirá la edad que mi amigo puede tener. Los años se separarán por un guion.

Entrada de ejemplo

```
2000 2024
2001 2024
2002 2020
2003 2024
0 0
```

Salida de ejemplo

```
23-24
22-23
17-18
20-21
```

H. DENTRO DEL LABERINTO

A Laber Into y a sus amigos les da por cada cosa...

Últimamente han cogido la afición de recorrer todos los laberintos de la zona. Juegan a ver quién encuentra primero la salida. Hasta ahora siempre ha ganado Laber, aunque no están muy convencidos de que lo haga de forma legal.



Después de mucho investigar, han conseguido los planos de varios laberintos y quieren averiguar si alguno no está bien construido, es decir, que se entre por un sitio y que se salga por otro. Si encuentran uno en el que no se pueda salir, piensan ir a visitarlo y si Laber también consigue salir de éste, ya no les quedará ninguna duda ¡hace trampas!

¿Les quieres echar una mano?

Entrada

El programa deberá leer, de la entrada estándar, múltiples casos de prueba, cada uno ocupando varias líneas.

En la primera línea aparecen dos números F, C, indicando el número de filas y columnas del laberinto ($2 \leq F, C \leq 100$)

A continuación, aparecen F líneas con C caracteres cada una indicando la situación del laberinto. Una X indica que hay una pared y no se puede pasar y un . indica que se puede avanzar por esa posición. Los avances han de ser siempre en horizontal o en vertical, nunca en diagonal.

Salida

Para cada caso de prueba el programa escribirá "SI" si se puede entrar por la entrada y salir por la salida y "NO" en caso contrario. Se va a suponer que la entrada está siempre en la esquina superior izquierda y la salida en la esquina inferior derecha.

Entrada de ejemplo

```
3 5
.....
....X
XXX..
3 5
.....
....X
XXXX.
5 5
.XXXX
..XXX
X..XX
XX..X
XXX..
5 5
.....
XXXX.
```

```
.....  
.XXXX  
.....
```

Salida de ejemplo

```
SI  
NO  
SI  
SI
```

I. EL AMIGO ALEMÁN

Nos hemos echado un amigo alemán que ha venido de Erasmus. No tiene ninguna pinta de alemán porque es moreno y no muy alto y además es tremendamente simpático. Se pasa el día hablando y contando anécdotas de su país.

Se llama Bartholomäus. Naturalmente lo llamamos Bartolo y estuvimos los primeros días riéndonos de su nombre y de los nombres en alemán.

Nos dejó de una pieza cuando nos dijo cuál era la palabra en alemán más complicada, o al menos la más larga: **Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz**, que significa “ley sobre la transferencia de las obligaciones de vigilancia del etiquetado de la carne de vacuno y la designación de los bovinos”. Nos contó que esta palabra es parte del título de una ley regional de Alemania, y el término se convirtió en la palabra oficial alemana más larga. Nadie tuvo valor para llevarle la contraria (ni para intentar pronunciarla).



Le dijimos que en español no tenemos nada parecido, así que nos dedicamos a buscar palabras con muchas consonantes seguidas y luego lo ampliamos también a vocales seguidas. Lo más que encontramos fueron palabras con 4 consonantes / vocales seguidas como por ejemplo, MONSTRUO, CONSTRUIR, LEIAIS, PREVEIAIS. Como resulta que Bartolo está estudiando Informática nos propuso hacer un programa para encontrar todas estas palabras en un texto, así que ... manos a la obra.

Entrada

La entrada comienza con el número de casos de prueba que se deberán procesar.

Cada caso ocupará una línea independiente y contendrá una frase de no más de 100 letras. Aunque suponga fallos ortográficos, por simplicidad ninguna vocal llevará tilde. Las palabras están separadas únicamente por un espacio.

Salida

Para cada caso de prueba se escribirá una línea con las palabras que tengan, al menos, 4 consonantes seguidas o 4 vocales seguidas. Entre cada dos palabras debe haber un espacio. En caso de no encontrar ninguna, se escribirá **Ninguna**. Tanto las consonantes como las vocales da igual que estén en mayúsculas o en minúsculas, pero a la hora de mostrarlas, deben aparecer en mayúsculas

Entrada de ejemplo

```
4
Me estoy leyendo un libro de monstruos
No es tan facil encontrar palabras con tantas consonantes seguidas en
español
Seguro que no construiais con el profesor clases abstractas en Java
porque no habia forma de entenderlas
Se me ha ocurrido una con 5 vocales y es entreoiais
```

Salida de ejemplo

MONSTRUOS
Ninguna
CONSTRUIAIS ABSTRACTAS
ENTREOIAIS

J. FELIZ 2024

De todas las felicitaciones que recibí el día de Año Nuevo hubo una que recuerdo especialmente, decía:

$$\text{Feliz } 2^3 + 3^3 + 4^3 + 5^3 + 6^3 + 7^3 + 8^3 + 9^3$$

Además de que me gustó mucho la felicitación por su ingenio, me hizo reflexionar sobre esta curiosidad e inventé los *números cúbicos*.

Un número es cúbico si es la suma de los cubos de una serie consecutiva de números.

La verdad es que no hay muchos, pero alguno hay. Por ejemplo, el año que viene también va a ser un año cúbico: $2025 = 1^3 + 2^3 + 3^3 + 4^3 + 5^3 + 6^3 + 7^3 + 8^3 + 9^3$. Para ver el siguiente ya habrá que esperar al 2197, que es igual a 13^3 .

¿Me ayudas a encontrar más números cúbicos?

Entrada

El programa deberá leer, de la entrada estándar, múltiples casos de prueba, cada uno ocupando una única línea.

En cada línea aparece un número indicando el año que queremos estudiar. Solo me interesa este milenio, así que los años van a estar entre el 2000 y el 2999.

La última línea es un cero que no debe procesarse.

Salida

Para cada caso de prueba, el programa escribirá una línea con la secuencia de números que, elevados al cubo y sumados, coinciden con el año. Como se ve en la salida de ejemplo, hay que escribir el primer número de la secuencia, un espacio, tres puntos, un espacio y el último número de la secuencia. En caso de que no exista dicha secuencia, deberá escribir NO CUBICO.

Entrada de ejemplo

```
2024
2025
2026
2197
0
```

Salida de ejemplo

```
2 ... 9
1 ... 9
NO CUBICO
13 ... 13
```



K. NÚMEROS HARSHAD

Este año, el 2024, es un número Harshad¹ y el anterior también y el siguiente también y el siguiente... ya no.

Bueno, pero ¿qué es un número Harshad? Se define como el número que es divisible por la suma de sus cifras. Así que decimos que el 2023 es Harshad porque es divisible por 7 ($2 + 0 + 2 + 3$). El 2024 también, porque es divisible por 8 ($2 + 0 + 2 + 4$). Lo mismo que el 2025, al ser divisible por 9 ($2 + 2 + 0 + 5$). El 2026 no es Harshad, porque no es divisible por 10 ($2 + 2 + 0 + 6$)



Hay bastantes números Harshad, así que muchos de ellos son consecutivos. Esto es justamente lo que queremos ver, cuántos números consecutivos son Harshad.

Entrada

El programa deberá leer, de la entrada estándar, múltiples casos de prueba, cada uno ocupando una única línea.

En cada línea aparecen dos números N, M ($1 \leq N < M \leq 10^6$) indicando el rango de números que queremos comprobar si son o no números Harshad.

La última línea son dos ceros que no deben procesarse.

Salida

Para cada caso de prueba, el programa escribirá la secuencia más larga de números Harshad consecutivos entre N y M . En caso de haber dos secuencias igual de largas, deberá escribir la compuesta por números más pequeños. En caso de que no haya ningún número Harshad entre N y M , deberá escribir "NO HAY NINGUNO"

Entrada de ejemplo

```
2000 2100
1 10
2011 2015
1000 2000
0 0
```

Salida de ejemplo

```
2022 2023 2024 2025
1 2 3 4 5 6 7 8 9 10
NO HAY NINGUNO
1014 1015 1016 1017
```

1 Los números Harshad fueron introducidos por primera vez en 1949 por el matemático indio Shri Dattatreya Ramachandra Kaprekar en un artículo publicado en el periódico "Scripta Mathematica"

L. QUEDANDO CON LA CUADRILLA

Después de no sé cuánto tiempo, hoy tenemos todos los amigos el día libre y por fin hemos quedado. Hemos estado super a gusto. El problema ha venido cuando hemos querido quedar para el próximo día.



- A ver, cuadrilla, ¿cuándo podemos quedar otra vez?
- Yo libro cada 3 días, así que dentro de 3 días, genial.
- Pues yo libro cada 4
- Je, je, yo cada 2 estoy libre
- Vaya rollo, a ver quién es el listo que calcula cuál es el próximo día que podemos quedar
- Eh, que os estáis olvidando que he empezado a estudiar Informática. Esto lo calculo yo en un periquete.

Entrada

La entrada comienza con el número de casos de prueba que se deberán procesar.

Cada caso es una línea en la que aparecen una serie de números separados por un espacio. El primero es un número N ($2 \leq N \leq 100$) que indica el número de amigos que somos en la cuadrilla. A continuación, aparecen N números indicando cada cuántos días libra cada uno de esos amigos. Me temo que alguno de los amigos no tiene muchas ganas de quedar y nos puede decir que libra dentro de un millón de días, así que los días pueden variar entre 2 y 1000000.

Salida

Para cada caso de prueba se escribirá dentro de cuántos días podemos volver a quedar todos juntos. En caso de que esta cifra sea mayor que 1000000, no vamos a quedar y escribiremos NUNCA

Entrada de ejemplo

```
4
3 3 4 2
2 3 2
2 3 3
3 2 3 1000000
```

Salida de ejemplo

```
12
6
3
NUNCA
```