

# Concurso de Programación Ada Byron 2026

## Regional Multisede

25 de marzo de 2026



## Cuadernillo de problemas



Organizado por



Universitat  
de les Illes Balears



*In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.*

**Ada Byron**

## Listado de problemas

A. Archivadores S.A.	3
B. Bombe, el Descifrador de Enigma	5
C. Oniso, ¡no te mojes!	7
D. Collarín colorado	9
E. Problema cinco?	11
F. Sa festa des Gambirots	13
G. Estos dioses están locos	15
H. Heathcliff en busca de Catherine	17
I. Por delante de los primeros	19
J. Premios narvalenses	21
K. Han cantado línea	23
L. Una ronda más y nos ponemos	25

Autores de los problemas:

- José María Buades Rubio
- Denis Molnar Ardelean
- Toni Fuentes Bauzá
- Miquel Oliver Alcayde
- Aina Maria Tur Serrano

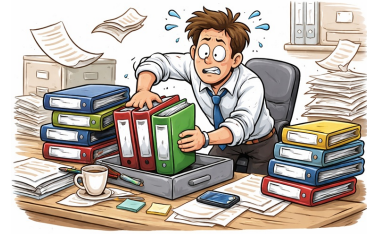


# ● A

## Archivadores S.A.

En el primer día de trabajo del becario Gumersindo en la empresa Archivadores S.A., recibe un WhatsApp del jefe. Urgentemente, debe bajar al archivo y ordenar los archivadores de los clientes. Si no lo hace, ese será su primer y último día.

Al bajar, ve un montón de archivadores. Cada uno tiene un número de cliente y una pegatina indicando la prioridad del cliente en la empresa (1 = más urgente, 5 = menos urgente). Sin embargo, los nervios y la preocupación por entregar el trabajo antes de que llegue el jefe provocan que algunos archivadores se rompan al abrirlos. Preocupado, Gumersindo decide cambiar su prioridad a 6 para moverlos al final.



Gumersindo decide cambiar su prioridad a 6 para moverlos al final.

Gumersindo debe ordenar los archivadores siguiendo estas reglas. Primero debe colocar los archivadores que no están rotos, ordenados por prioridad (de más urgente a menos). Si dos archivadores tienen la misma prioridad, se ordenan por número de cliente (de menor a mayor). Los archivadores rotos deben colocarse al final, ordenados siguiendo la misma lógica.

### Entrada

La entrada comienza con un entero  $t$  que indica el número de casos de prueba. Cada caso de prueba se describe en una línea que comienza con un entero  $n$  ( $1 \leq n \leq 10^4$ ), que indica el número de archivadores. A continuación aparecen  $n$  pares de números. Cada par representa un archivador mediante: el número de cliente  $w$  ( $1 \leq w \leq 10^5$ ) y su prioridad  $p$  (entero entre 1 y 6), separados por espacios. Las prioridades del 1 al 5 indican archivadores en buen estado (siendo 1 la más urgente), mientras que la prioridad 6 indica que el archivador está roto.

### Salida

Para cada caso de prueba, se debe imprimir primero la lista de archivadores en buen estado, ordenados según prioridad y número de cliente, y después los archivadores rotos, indicando que están rotos.

### Entrada de ejemplo

```
4
7 120 3 150 1 200 2 300 6 500 6 180 1 250 3
4 10 1 20 6 30 1 40 6
5 5 3 2 1 8 6 1 2 7 6
11 90 2 45 1 300 6 210 3 500 6 10 1 75 2 600 6 33 3 120 1 450 2
```

### Salida de ejemplo

```
150 1 180 1 200 2 120 3 250 3 300 ROTO 500 ROTO
10 1 30 1 20 ROTO 40 ROTO
2 1 1 2 5 3 7 ROTO 8 ROTO
10 1 45 1 120 1 75 2 90 2 450 2 33 3 210 3 300 ROTO 500 ROTO 600 ROTO
```

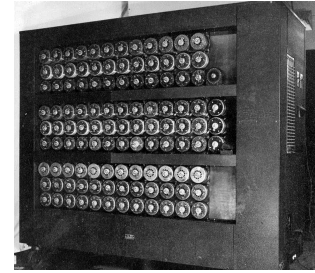


## ● B

# Bombe, el Descifrador de Enigma

En septiembre de 1939, Gran Bretaña declaró la guerra a Alemania. En ese mismo momento, el matemático Alan Turing era reclutado por el ejército británico como criptólogo en la base militar de Bletchley Park. ¿Su misión? Revelar el sistema de cifrado de la máquina Enigma, utilizada por los nazis para proteger sus comunicaciones secretas.

La máquina funcionaba de forma similar a una máquina de escribir, pero tenía un complejo mecanismo interno formado por varios rotores que, al pulsar en una letra, giraban y la sustituían por otra. Para encriptar el mensaje, cada rotor se colocaba inicialmente en una posición determinada. Tras cada pulsación, los rotores avanzaban, lo que cambiaba continuamente el sistema de cifrado. Este mecanismo generaba millones de combinaciones posibles.



Turing y su equipo trabajaban contrarreloj, conscientes de que cada mensaje descifrado podía salvar miles de vidas, aunque la gran cantidad de combinaciones hacían imposible cumplir con la misión. El tiempo apremiaba, así que Turing pensó que la mejor forma de derrotar a una máquina era con otra máquina.

A finales de 1939, lo consiguieron. Crearon Bombe, una máquina capaz de descartar gran parte de las combinaciones imposibles y acelerar enormemente el proceso de descifrado. Ahora, vuestra misión es continuar su trabajo: utilizad la combinación de rotores que proporciona Bombe y tratad de descifrar el mensaje.

Esta máquina solo funciona con letras mayúsculas del alfabeto anglosajón, y para simplificar el problema, vuestra máquina tiene un único rotor. Este rotor representa un desplazamiento dentro del alfabeto. Cuando se descifra el mensaje, cada letra de la frase cifrada se desplaza varias posiciones hacia atrás en el abecedario, según el valor del rotor en ese momento. Después de procesar una letra, el rotor avanza una posición. Esto se puede expresar como:  $rotor = (rotor + 1) \text{ mód } 26$ , de modo que, después de la Z, el rotor vuelve a empezar desde el principio del alfabeto.

Esta máquina solo funciona con letras mayúsculas del alfabeto anglosajón, y para simplificar el problema, vuestra máquina tiene un único rotor. Este rotor representa un desplazamiento dentro del alfabeto. Cuando se descifra el mensaje, cada letra de la frase cifrada se desplaza varias posiciones hacia atrás en el abecedario, según el valor del rotor en ese momento. Después de procesar una letra, el rotor avanza una posición. Esto se puede expresar como:  $rotor = (rotor + 1) \text{ mód } 26$ , de modo que, después de la Z, el rotor vuelve a empezar desde el principio del alfabeto.

## Entrada

La entrada del programa consiste en dos números: el primero indica la cantidad de frases a descifrar y el segundo indica la posición inicial del rotor, que puede ser un entero del 1 al 26. Después, aparecen cada una de las frases cifradas. Cada frase puede tener una longitud máxima de 128 caracteres y se descifran con el mismo rotor inicial.

## Salida

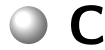
La salida devuelve las frases descifradas.

## Entrada de ejemplo

```
4 10
RZYN AJDUG
FLYBG P WRFTL ZOQC BOOEXVXU
DCQF HGYJLXM OEDPDS UTDKFT AZRQZ QA IC JIAZUG
OYTBFPRLWGU CWYCHS EGVGNLYIMY PX ZSCIRBX
```

## Salida de ejemplo

```
HOLA MUNDO
VAMOS A GANAR ESTE CONCURSO
TRES TRISTES TIGRES TRAGAN TRIGO EN UN TRIGAL
ENHORABUENA HABEIS DESCIFRADO EL MENSAJE
```



# Oniso, ¡no te mojes!

Oniso es una oniscidea (bicho bola), que tiene una forma muy curiosa de jugar. Se hace una bolita y se deja caer por las pendientes de su jardín. Pero no le gusta rodar hasta el agua porque su madre le castigará sin respirar durante cinco minutos. Su jardín es un tanto especial, ya que a diferencia de nuestro mundo, el suyo es bidimensional, solo tiene anchura (eje  $x$ ) y altura (eje  $y$ ), y no tiene profundidad.



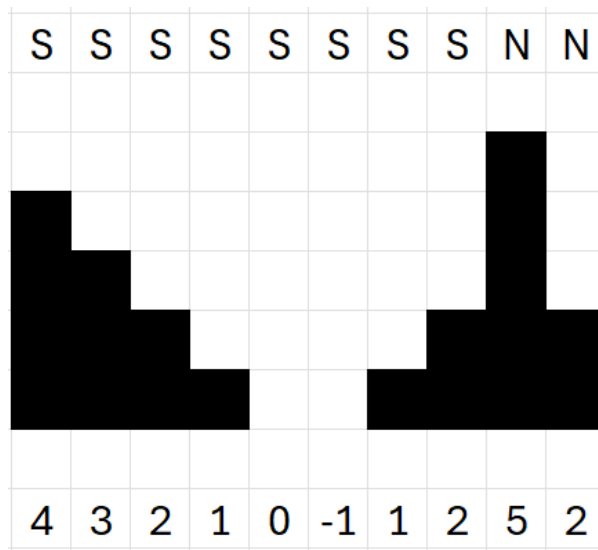
Como hemos dicho, Oniso vive en un mundo bidimensional. Si se hace bola en una posición  $x$  rueda hacia  $x + 1$  si la altura en  $x + 1$  es menor que en  $x$  y la altura en  $x - 1$  es mayor que en  $x$ . De forma simétrica, rodará desde la posición  $x$  hacia la posición  $x - 1$  si la altura en ese punto es menor y en  $x + 1$  es mayor. En el resto de casos no rueda. Oniso rueda hasta que llega a un punto que no puede rodar.

El agua está en todas las posiciones que su altura es igual a cero o menor.

Tenemos que ayudarle indicando desde cuántos puntos va a acabar mojándose (pasará por agua). Hay que tener en cuenta que fuera del jardín la altura es siempre una unidad mayor que cualquier altura del jardín.

En la siguiente figura se muestra el jardín de Oniso. Abajo aparecen las alturas, y arriba (S) si acaba mojándose y (N) si no acaba mojándose.

Tenemos que decirle desde cuántas coordenadas terminará en el agua. En este caso desde las 8 posiciones que tiene encima la letra S.



## Entrada

La entrada comienza con un número  $t$  que indica el número de jardines que hay que calcular ( $t \leq 2 \cdot 10^4$ ). A continuación, para cada jardín aparece un número  $w$  que es la anchura del jardín ( $1 \leq w \leq 10^5$ ), y en la siguiente línea las alturas de las  $w$  posiciones:  $a_1, a_2, \dots, a_w$ . Se cumple que  $-10^9 \leq a_i \leq 10^9$ . En cada caso de prueba, el total de número de alturas no superará  $10^6$ .

## Salida

Para cada caso de prueba tiene que indicar el número de posiciones en las que acabará mojándose.

### Entrada de ejemplo

```
2  
10  
4 3 2 1 0 -1 1 2 5 2  
6  
0 1 1 1 0 1
```

### Salida de ejemplo

```
8  
3
```

## ● D

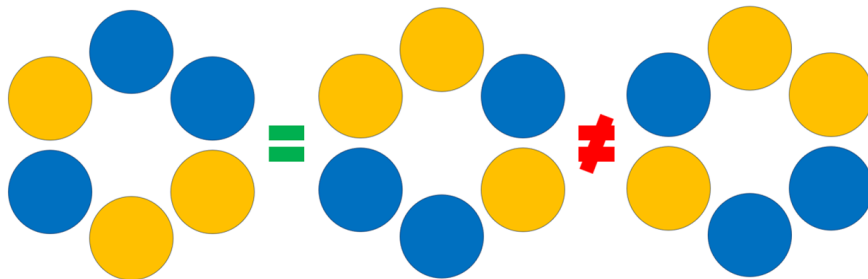
# Collarín colorado

Érase una vez un joven bisutero llamado Matías que soñaba con alcanzar la estabilidad económica vendiendo collares. Sus habilidades eran insuperables en todo el reino; sin embargo, necesitaba ayuda para determinar si su negocio podría llegar a ser rentable.

Matías fabrica collares utilizando cuentas de dos colores distintos. Para cada collar utiliza exactamente una cantidad fija de cuentas de cada color. Como auténtico maestro de su oficio, se niega a vender dos collares idénticos, por lo que quiere saber cuántos collares diferentes puede crear con las cuentas disponibles.

Dos collares se consideran iguales si, al rotarlos, se obtiene la misma secuencia de colores en el mismo orden. Es decir, las rotaciones de un collar no generan un diseño diferente. Sin embargo, los collares de Matías son verdaderas obras de arte y está mal visto llevarlos del revés: las reflexiones (efecto espejo) no se consideran equivalentes.

En la siguiente imagen se muestra un mismo collar representado mediante dos rotaciones distintas, así como otro collar diferente que, aunque similar, no es igual.



### Entrada

La primera línea contiene un número entero  $t$ , que indica el número de casos de prueba.

Cada uno de los siguientes  $t$  casos de prueba consiste en dos números enteros no negativos que representan la cantidad de cuentas disponibles de cada uno de los dos colores. La suma de ambos números es al menos 1 y no supera 20.

### Salida

Para cada caso de prueba se debe imprimir en una línea el número total de collares diferentes que se pueden construir utilizando exactamente todas las cuentas disponibles.

### Entrada de ejemplo

```
2
3 3
1 1
```

### Salida de ejemplo

```
4
1
```



# ● E

## Problema cinco?

“Acércate que te brinco!” dijo Juan que llevaba obsesionado con esa rima igual que el resto de compañeros de clase estos últimos meses.

Su obsesión llega a tal nivel que ha decidido que a partir de ahora va a intentar siempre que mire la tele intentar poner un valor de volumen que tenga esa rima por si alguien lo revisa en voz alta.

Para ello, dado el volumen actual de la tele quiere saber a qué valor debería dejarlo para que acabe en 5. Ten en cuenta que números acabados en 15 no le sirven porque “quince” no rima con “cinco”.

Pero tiene un problema, el botón de bajar el volumen no le funciona, así que solo puede subir el volumen.



### Entrada

La entrada está formada por un único número  $V$  ( $1 \leq V \leq 1005$ ) indicando el volumen actual del televisor.

### Salida

La salida es el valor del volumen al cual debería dejar el televisor.

#### Entradas de ejemplo

7

5

26

554

#### Salidas de ejemplo

25

5

35

555





## Sa festa des Gambirots

La *colla casteller* universitaria “Gambirots de la UIB” está preparando una importante actuación para celebrar su segundo aniversario, y, para la ocasión, el *casteller* Pinyol ha decidido grabar una versión orquestal de su himno. Por suerte, hay bastantes integrantes que están especializados en el mundo de la música y que saben tocar un instrumento o incluso varios, por lo que no será necesario contratar a ningún músico. Sin embargo, el compositor encargado de la orquestación se emocionó poniendo muchas partes a los instrumentos, y como no hay muchos músicos, probablemente no podrán tocar toda la pieza de una sola vez, de modo que tendrán que grabarla en varias tomas por separado, para luego juntarlas.



Así, por ejemplo, si tenemos una violinista, una violista y un chelista en la *colla*, y queremos grabar una versión del himno para cuarteto de cuerda (con dos violines, una viola y un violonchelo), entonces tendremos que hacer dos grabaciones: la primera, con los tres músicos a la vez; y la segunda, para la parte de violín que queda por grabar.

Si, por otra parte, suponemos que la violinista y el chelista saben además tocar la gralla<sup>1</sup>, y queremos tocar para un sexteto con dos violines, viola, violonchelo y dos grallas, entonces tendremos que realizar tres tomas para grabar toda la pieza:

	Intérprete 1	Intérprete 2	Intérprete 3
Toma 1	Violín	Viola	Violonchelo
Toma 2	Violín	–	Gralla
Toma 3	Gralla	–	–

Los estudios de grabación suelen costar bastante dinero y la *colla* no dispone de demasiado presupuesto para ello. Por eso, Blat, el nuevo jefe de tesorería, te ha pedido cuál es el número mínimo de tomas que tendrán que hacer para grabar toda la obra.

### Entrada

La entrada estará formada por distintos casos de prueba.

Cada caso de prueba empieza con una línea con dos números que indican el número  $T$  de tipos de instrumentos distintos que aparecen en la partitura (hasta 70), y el número  $M$  de músicos que hay en la *colla* (entre 1 y 70). A continuación, aparecerá una línea con  $T$  números, indicando el número de partes que deberán grabarse para cada instrumento (entre 1 y  $10^9$ ).

Cada una de las siguientes  $M$  líneas empieza con el número  $n_i$  de instrumentos distintos que sabe tocar el  $i$ -ésimo músico, seguido de  $n_i$  números con los identificadores de tales instrumentos (entre 1 y  $T$ ). Hay que tener en cuenta que, en una misma grabación, cada músico puede tocar una parte de uno de sus instrumentos o no tocar, pero no puede tocar a la vez dos partes distintas de un mismo instrumento o de instrumentos distintos.

La entrada termina con una línea con dos ceros que no debe procesarse.

### Salida

Por cada caso de prueba se escribirá una línea con el número mínimo posible de grabaciones necesarias para poder grabar todas las partes del himno de la *colla*. Si es imposible grabar la partitura con los músicos que hay en la *colla*, se escribirá en su lugar IMPOSIBLE en una línea.

<sup>1</sup>Instrumento de viento madera tradicional catalán, ampliamente utilizado en el mundo *casteller*.

### Entrada de ejemplo

```
3 3
2 1 1
1 1
1 2
1 3
4 3
2 1 1 2
2 1 4
1 2
2 3 4
3 1
2 1 1
1 1
0 0
```

### Salida de ejemplo

```
2
3
IMPOSIBLE
```

# ● G

## Estos dioses están locos

Llogurcito es un estudiante de quinto de historia antigua. Está estudiando la mitología griega, que es muy extensa y extraña. Para nosotros cada ser humano es descendiente de un padre y una madre. Pero para ellos esto sería demasiado sencillo. En el universo mitológico sucede que salen seres a partir de un solo ser, de dos, de tres, o de los que haga falta. Incluso algunos seres como Caos han sido engendrados de la nada. Pero en ningún caso sucede que un personaje es engendrado por el mismo.



Llogurcito, aunque le cuesta, acepta que esto pueda suceder, pero no va a aceptar que un personaje de la mitología sea descendiente de un propio descendiente suyo, ¡eso sí que no!

Tenemos que indicarle si la configuración del árbol genealógico es posible.

### Entrada

La entrada comienza con un número  $t$  que indica el número de árboles genealógicos a tratar. Cada árbol comienza con el número de personajes  $N$ . A continuación, para cada personaje  $i$  ( $1 \leq i \leq N$ ) tenemos  $M_i$  el número de ascendientes que tiene el personaje  $i$ , seguido de  $M_i$  números  $a_1, a_2, \dots, a_{M_i}$ , los índices de sus antecesores ( $1 \leq a_j \leq N$  y  $a_j \neq i$ ). Entre todos los casos de prueba nunca se superará los  $10^4$  personajes.

### Salida

Para cada caso de prueba tiene que indicar si es posible (SI) o no es posible (NO).

### Entradas de ejemplo

```
2
4
0
1 1
3 1 2 4
2 1 2
5
1 2
1 3
2 1 2
0
0
```

### Salidas de ejemplo

```
SI
NO
```





# Heathcliff en busca de Catherine

Una terrible noticia llega a Cumbres Borrascosas desde la casa de los Linton: Catherine está gravemente enferma.

Bajo la lluvia y el viento del páramo, Heathcliff debe salir de inmediato para llegar hasta ella, ya que si no llega a tiempo, teme que pueda perderla para siempre. Pero el camino está lleno de senderos que lo llevan por colinas, bosques y ríos. A lo largo del recorrido hay claros en el páramo donde varios caminos se cruzan. Heathcliff debe decidir: ¿deberá subir la colina primero o atravesar el río? Cada sendero tiene un tiempo distinto, y cada minuto cuenta.

Ayuda a Heathcliff a elegir el camino más rápido para llegar a Catherine y no perder ni un instante.

## Entrada

La entrada está compuesta por diversos casos de prueba, ocupando cada uno de ellos varias líneas.

La primera línea contiene tres enteros:  $N$ ,  $M$  y  $S$ .  $N$  indica el número de claros ( $2 \leq N \leq 10^5$ ),  $M$  indica el número de senderos ( $N - 1 \leq M \leq 6N$ ) y  $S$ , el claro desde donde comienza Heathcliff su camino ( $1 \leq S \leq 10^5$ ). La siguiente línea contiene un entero que indica el lugar de destino ( $1 \leq D \leq 10^5$ ), la casa de los Linton, donde se encuentra Catherine. Las siguientes  $M$  líneas describen los senderos. Cada una contiene tres enteros:  $U$ ,  $V$  y  $W$ .  $U$  y  $V$  indican los claros conectados por un camino ( $1 \leq u, v \leq N$ ) y  $W$  indica el tiempo que tarda en recorrer ese sendero ( $1 \leq W \leq 100$ ).

Se garantiza que para todos los casos el camino mínimo existe y es único.

## Salida

Para cada caso de prueba se escribirá una línea con el camino que debe realizar Heathcliff y el tiempo total que tarda en recorrerlo.

## Entrada de ejemplo

```
5 6 1
5
1 2 10
1 3 5
2 3 2
2 4 1
3 4 9
4 5 4
6 8 2
6
1 2 3
2 3 4
2 4 2
3 5 6
4 5 1
4 6 7
5 6 2
1 3 5
```

## Salida de ejemplo

```
1 3 2 4 5 -> 12
2 4 5 6 -> 5
```





# Por delante de los primeros

A diferencia de otros concursos de programación como el *AdaByron* que estáis concursando ahora mismo, en la Olimpiada Informática cada envío puede dar un número variable de puntos. Esta medida pretende evitar situaciones de empate, por lo que los envíos erróneos, o el tiempo que se tarda en resolver un problema, pasan a ser preocupaciones menores para los concursantes.



Para motivar a sus alumnos a participar, José María, un profesor de instituto, aseguró que si alguien quedaba *estrictamente* por delante del primer puesto, entonces los invitaría a todos a cenar. Tras darse cuenta de que eso era matemáticamente imposible (dado que quedar en primer puesto no cuenta como quedar por delante de los primeros), al final se conformó con que alguien quedase por delante de la  $k$ -ésima posición. Si alguno de sus estudiantes lo lograba, prometió invitarlos a todos a una deliciosa merienda en la cafetería de la facultad. Al fin y al cabo, el profesor estará contento de su duro trabajo, siempre y cuando no terminen por detrás de los últimos.

## Entrada

La entrada empieza con una línea indicando el número de casos de prueba que vendrán a continuación.

Cada caso de prueba empieza con una línea con tres números que indican el número  $n$  de participantes que hay en el concurso (hasta  $10^5$ ), el número total de envíos  $m$  que se producen a lo largo del concurso (hasta  $10^5$ ) y la posición  $k$  en la que se quiere estar durante el tiempo de concurso (entre 1 y  $n$ ).

A continuación, vienen  $m$  líneas con la información de todos los envíos, ordenados según el instante en el que se producen. Cada una de esas líneas contiene dos enteros, indicando el identificador del concursante que ha realizado ese envío (entre 1 y  $n$ ), y la cantidad de puntos conseguidos (entre 0 y  $10^9$ ).

## Salida

Por cada caso de prueba se escribirá una línea con  $m$  números separados por un espacio. Cada número indica, tras cada envío, la puntuación que tendrá el concursante que ocupe la  $k$ -ésima posición.

## Entrada de ejemplo

```
2
3 4 2
1 5
2 4
3 7
1 2
3 3 3
1 5
2 4
3 7
```

## Salida de ejemplo

```
0 4 5 7
0 0 4
```



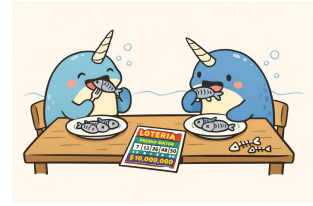


# Premios narvalenses

La Organización de Narvales Canadienses y Estadounidenses (ONCE) ha organizado una lotería con un premio de hasta 30 bacalaos de cada número premiado. Cada boleto contiene un número de hasta 30 dígitos.

Como este año han pescado muchos bacalaos, han decidido que se ha sortearían varios números ganadores (hasta  $10^5$ ) y el premio de cada boleto jugado se calcularía como la suma de las longitudes de los prefijos que coinciden con los diferentes números ganadores.

Tu tarea es, dada la lista de boletos y los números ganadores, determinar cuántos bacalaos tendrán que preparar para dar todos los premios. Además, los narvales canadienses nos aseguran que como máximo han repartido  $10^5$  boletos.



## Entrada

La entrada consiste en un primer número  $m$  ( $1 \leq m \leq 10^5$ ) con el total de boletos ganadores seguido de  $m$  líneas con el valor de estos boletos. Los boletos premiados pueden repetirse y no necesariamente habrá alguien con el mismo valor que un boleto premiado. Seguidamente viene el número  $n$  ( $1 \leq n \leq 10^5$ ) de boletos vendidos, con  $n$  líneas de los valores de esos boletos vendidos.

Todos los boletos contienen la misma cantidad de dígitos y, estos dígitos, siempre son valores primos entre 1 y 9.

La entrada acaba con un caso con 0 boletos ganadores.

## Salida

La salida consiste en el número de pescados que se lleva cada uno. Al final de cada caso, tiene que aparecer el total de bacalaos que necesitaran para poder dar todos los premios y deberá estar separado del siguiente caso de prueba por ---.

## Entrada de ejemplo

```
4
223357
775532
223357
235757
5
223357
775532
235723
225537
227335
0
```

## Salida de ejemplo

13
6
6
5
5
35
---

## Aclaración

Si juegas con el boleto 235, y los premiados son 232, 235 y 225 obtienes 6 bacalaos. Dos bacalaos por 232, tres por 235 y uno por 225.

# ● K

## Han cantado línea

*Abrazacharcos* es una foca monje del mediterráneo que le encanta hacer amigos y amigas. El pasado verano lo pasó en Cabrera con su prima *Saltaespuma*, y en una noche de luna nueva conocieron a *ForatNegre*, un peculiar ser de otro planeta. *ForatNegre* les contó que en su planeta los años duran  $7 \cdot 10^6$  días. Como los años son muy largos celebran semanas con una gran fiesta, y tienen los mismos días de la semana: lunes, martes, miércoles, jueves, viernes, sábado y domingo. Además, todos los años el primer día del año es jueves.



Sabiendo el día del año que han nacido todos los familiares de *ForatNegre* quieren conocer cuántos días de la semana diferentes tienen fiesta. Y si tienen un pleno, los siete días de la semana tienen fiesta, cantan línea.

### Entrada

La entrada comienza con un número  $t$  que indica el número de casos de prueba que hay que calcular ( $t \leq 1000$ ). A continuación, para cada caso de prueba tenemos  $N$ , el número de familiares de *ForatNegre* ( $1 \leq N \leq 5000$ ), y en la siguiente línea las  $N$  fechas de nacimiento de estos:  $a_1, a_2, \dots, a_N$ , donde  $1 \leq a_i \leq 7 \cdot 10^6$ .

### Salida

Para cada caso de prueba tiene que indicar el número de días de la semana diferentes en los que tienen fiesta, si han nacido en los siete días tiene que gritar LINEA.

### Entrada de ejemplo

```
3
3
1 1 1
3
4 11 18
7
418 5405 198 3835 6135 1771 347
```

### Salida de ejemplo

```
1
1
LINEA
```



# ● L

## Una ronda más y nos ponemos

Desde que se popularizó un conocido juego de cartas por equipos en su universidad, Juan aprovecha cada rato libre que tiene para buscar a más personas con las que jugar. Como resultado, siempre acaban los mismos cuatro pasando las horas en la cafetería de la facultad, o quedando cualquier domingo por la tarde para echarse una partida en algún bar.

Un día, tras jugar algunas rondas amistosas contra el equipo *Envido de Fenwick*, decidieron empezar a contabilizar los puntos para darle emoción, de manera que ganase el primer equipo en acumular 24 puntos. Sin embargo, la cosa se torció y Juan acabó perdiendo varias partidas seguidas. No obstante, esto le dejó pensando... ¿Habrían podido ganar si hubiesen empezado a contar los puntos en otro momento?



### Entrada

La entrada estará formada por distintos casos de prueba.

Cada caso de prueba empieza con una línea con dos números, indicando el número de rondas jugadas en total (hasta  $10^6$ ) y el número de puntos necesarios para ganar una partida (hasta  $10^9$ ).

A continuación, viene una línea con un número entre  $-10^9$  y  $10^9$  por cada ronda. Un entero positivo  $x > 0$  indica una cantidad  $|x|$  de puntos ganados en esa ronda. En cambio, un número negativo  $x < 0$  indica que el equipo rival ha ganado  $|x|$  puntos. Finalmente, un cero indica que no se han otorgado puntos a ningún equipo durante esa ronda.

Se garantiza que entre todos los casos de prueba no habrá más que un total de  $10^6$  rondas.

### Salida

Por cada caso de prueba, se escribirá una línea con el número máximo de partidas seguidas que se podrían haber ganado, si se pudiese haber escogido el momento en el que se empezaron a contabilizar los puntos de la primera partida. Ten en cuenta que cada vez que termina una partida y empieza la siguiente justo después, los marcadores de puntos se vuelven a poner a cero.

### Entrada de ejemplo

```
12 24
9 6 3 9 -24 9 26 6 -3 9 3 6
17 47
43 44 32 0 -10 4 37 28 -13 18 26 -22 25 -1 -33 23 35
```

### Salida de ejemplo

```
2
4
```