

Universidad Mayor de San Andrés - Carrera de Informática

## International Collegiate Programming Contest



**Club de Programación  
Competitiva  
{UMSA}**

## Competencia Interna Segunda División 2026

1 de agosto de 2026

*Las paginas están numeradas desde el 1 al 26 (sin contar caratula e indice).*

### Setters:

*Ajhuacho Tarquino, Luis David  
Castillo Uscamayta, Marco Antonio  
Condori Mamani, Eliseo Eliezer  
Los maquinolas: The last dance  
Vargas Garcia, Kelvin Bohn Davisson*

### Judges/Organizers:

*Carvajal Yucra, Oscar Gauss  
Flores Condori, Romer  
Loza Ramirez, Samuel  
Quispe Mamani, Miguel Angel  
Troche Venegas Rolando*

### Sponsors:



## Índice

|                                              |    |
|----------------------------------------------|----|
| A. Sincronizacion Espacial en Code XD        | 1  |
| B. El Illimani no tiene nada en contra de mi | 3  |
| C. Monte Carlmore                            | 5  |
| D. Destino Final                             | 7  |
| E. Día de la Amistad                         | 10 |
| F. F maestro                                 | 12 |
| G. Jean Grey                                 | 13 |
| H. La promesa del Capitán Cornelio           | 15 |
| I. Los cuadernos aesthetic de Leif           | 17 |
| J. Suertudo                                  | 19 |
| K. Bron                                      | 20 |
| L. Un Jardín Armonioso                       | 22 |
| M. El robo final                             | 24 |

# Problema A

## Sincronizacion Espacial en Code XD

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Kelvin Bohn Davisson Vargas Garcia*

Eres el ingeniero en jefe de la estación espacial **Code XD**. Se te ha entregado un arreglo  $a$  que consta de  $n$  números enteros positivos, los cuales representan los niveles de energía de los módulos de soporte vital de la nave.

Inicialmente, el reloj de calibración cuántica de la nave marca  $x = 0$ . Durante un movimiento (que equivale a un pulso temporal), puedes realizar una de las dos operaciones siguientes:

1. Elegir exactamente un módulo  $i$  (desde 1 hasta  $n$ ) e incrementar su energía  $a_i$  en  $x$  ( $a_i := a_i + x$ ), y luego el reloj  $x$  avanza en 1 ( $x := x + 1$ ).
2. Simplemente dejar que el reloj  $x$  avance en 1 sin calibrar ningún módulo ( $x := x + 1$ ).

La primera operación (calibrar un módulo) se puede aplicar **no más de una vez** a cada módulo  $i$  (desde 1 hasta  $n$ ).

Para que la nave pueda entrar en el hiperespacio, todos sus módulos deben estar sincronizados. Tu tarea es encontrar el **número mínimo de movimientos** requeridos para obtener un arreglo de energía donde cada uno de sus elementos sea **divisible por 10**

### Entrada

La primera línea contiene un número entero  $n$  ( $1 \leq n \leq 100$ ) — la cantidad de módulos de energía en el arreglo  $a$ . La segunda línea del caso de prueba contiene  $n$  números enteros  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 9$ ), donde  $a_i$  es la energía inicial del  $i$ -ésimo módulo.

### Salida

Imprime el número mínimo de movimientos requeridos para obtener un arreglo en el que cada elemento sea divisible por 10.

### Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 4<br>6 5 7 0         | 6                   |

| Ejemplo de entrada 2      | Ejemplo de salida 2 |
|---------------------------|---------------------|
| 10<br>8 2 8 8 7 1 7 2 5 4 | 23                  |

## Nota

Considera el primer caso de prueba del ejemplo:

- $x = 0$ ,  $a = [6, 5, 7, 0]$ . Simplemente incrementamos 3 veces  $x$  (Operación 2).
- $x = 3$ ,  $a = [6, 5, 7, 0]$ . Sumamos  $x$  al tercer módulo e incrementamos  $x$  (Operación 1).
- $x = 4$ ,  $a = [6, 5, 10, 0]$ . Sumamos  $x$  al primer módulo e incrementamos  $x$  (Operación 1).
- $x = 5$ ,  $a = [10, 5, 10, 0]$ . Sumamos  $x$  al segundo módulo e incrementamos  $x$  (Operación 1).
- $x = 6$ ,  $a = [10, 5, 10, 0]$ . Hemos obtenido el arreglo requerido (todos son divisibles por 10).

Ten en cuenta que no puedes sumar la señal  $x$  al mismo módulo de energía más de una vez a lo largo del proceso.

# Problema B

## El Illimani no tiene nada en contra de mi

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Marco Reus*

Reus es profesor en la Universidad del Illimani. (.<sup>El</sup> Illimani no tiene nada en contra de mi, el Illimani no tiene nada en contra de mi...")

Los asientos de su aula están organizados en dos filas, con  $N$  asientos en cada una.

Reus está enseñando a  $A + B + C$  estudiantes y quiere sentar a la mayor cantidad posible.

Él sabe que:

- El grupo de  $A$  estudiantes solo quiere sentarse en la fila 1.
- El grupo de  $B$  estudiantes solo quiere sentarse en la fila 2.
- El grupo de  $C$  estudiantes no tiene preferencia y puede sentarse en cualquiera de las dos filas.

Cada asiento puede ser ocupado por un solo estudiante y, si un estudiante es sentado, debe respetarse su preferencia.

¿Cuál es el máximo número de estudiantes que Reus puede sentar?

### Entrada

La primera línea contiene un entero  $t$  ( $1 \leq t \leq 10^4$ ), que representa el número de casos de prueba.

Cada caso de prueba contiene cuatro enteros  $N$ ,  $A$ ,  $B$  y  $C$  ( $1 \leq N, A, B, C \leq 10^8$ ).

### Salida

Para cada caso de prueba, imprime el máximo número de estudiantes que pueden ser sentados.

### Ejemplos

| Ejemplo de entrada 1                                             | Ejemplo de salida 1      |
|------------------------------------------------------------------|--------------------------|
| 5<br>10 5 5 10<br>3 6 1 1<br>15 14 12 4<br>1 1 1 1<br>420 6 9 69 | 20<br>5<br>30<br>2<br>84 |

# Problema C

## Monte Carlmore

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*  
*Autor: Romer Flores Condori*



Figura 1: Monte Carlmore (Los Simpson T13/E10)

Una maravillosa tarde de verano, Lenny habia tallado el monte Carlmore, hoy se propone volver, pero tiene un pequeño problema, no recuerda donde es la cima del monte, entonces decide usar su mapa, el cual consta de tres numeros  $a, b$  y  $c$ , la posicion actual de Lenny es  $b$ .

- Lenny se encuentra en la cima si  $a < b > c$ .
- Lenny se encuentra aun subiendo el monte si  $a < b < c$ .

Ayuda a Lenny a saber si aun esta subiendo o ya llego a la cima.

### Entrada

La primera linea consta de un entero  $t$  ( $1 \leq t \leq 1000$ ) el cual representa la cantidad de casos, cada caso consta de tres numeros enteros  $a, b, c$  ( $1 \leq a, b, c \leq 1000$ )

### Salida

Por cada caso de prueba, si Lenny se encuentra en la cima debes imprimir la palabra "CIMA" (sin comillas), si Lenny se encuentra aun subiendo debes imprimir "SUBIENDO" (sin comillas) y si no se cumple ninguna de estas condiciones debes imprimir "NINGUNO" (sin comillas).

### Ejemplos

| Ejemplo de entrada 1         | Ejemplo de salida 1         |
|------------------------------|-----------------------------|
| 3<br>4 5 6<br>1 4 2<br>1 4 4 | SUBIENDO<br>CIMA<br>NINGUNO |

# Problema D

## Destino Final

*Tiempo Límite: 3 seg, Memoria Límite: 256 MB*

*Autor: Eliseo Eliezer Condori Mamani*

Una curiosa maldición relacionada con la muerte está ocurriendo. Una chica llamada Iris tuvo una premonición sobre un accidente sangriento en una torre. Ella advirtió a todas las personas presentes y evitó la muerte de muchas de ellas. Sin embargo, años después, la muerte comenzó a perseguir a cada una de las personas que estuvieron involucradas en aquel accidente, así como a todas sus generaciones descendientes (hijos, nietos, etc.).

En la actualidad, la muerte ya ha reclamado prácticamente todas las vidas relacionadas con aquella tragedia, pero existe una forma de evitar la maldición. William Bludworth, un hombre misterioso y adicto a los cameos, conocía un caso especial: cuando una persona logra escapar de la muerte, es decir, muere temporalmente y luego es reanimada, satisface temporalmente las exigencias de la Muerte.

Inspirado por la historia de la película *Destino Final*, el autor de este problema se preguntó lo siguiente: si una persona de la familia de Iris logra evitar la muerte, ¿cuántas personas de sus descendientes quedarían salvadas temporalmente? Al evitar la cadena de muertes, también se salvarían sus hijos, nietos y demás descendientes.

Para modelar el problema, el autor representa a los miembros de la familia mediante números del 1 al  $n$ . Iris está representada por el número 1, y todos los demás miembros descienden de algún antepasado dentro de la familia de Iris.

El autor desea responder dos tipos de preguntas:

- Si una persona  $x$  evita la muerte, ¿cuántas generaciones de descendientes se salvarían?
- Si una persona  $x$  evita la muerte, ¿cuántas personas se salvarían en total, incluyéndose a sí misma?

Como tiene interés en varios miembros de la familia, realizará  $q$  consultas.

### Entrada

La primera línea contiene un entero  $n$  ( $1 \leq n \leq 10^5$ ), el número de miembros de la familia de Iris.

Las siguientes  $n-1$  líneas contienen dos enteros  $a$  y  $b$ , indicando que existe una relación familiar entre  $a$  y  $b$ .

La siguiente línea contiene un entero  $q$  ( $1 \leq q \leq 2 * n$ ), el número de consultas.

Cada una de las siguientes  $q$  líneas tiene el formato  $t \ x$ , donde:

- $t$  es el tipo de consulta ( $t \in \{A, B\}$ ):

- A: preguntar cuántas generaciones de descendientes de  $x$  se salvarían.
- B: preguntar cuántas personas se salvarían en total, incluyendo a  $x$ .
- $x$  es la persona que logra evitar la muerte.

## Salida

Imprime  $q$  líneas.

Para cada consulta:

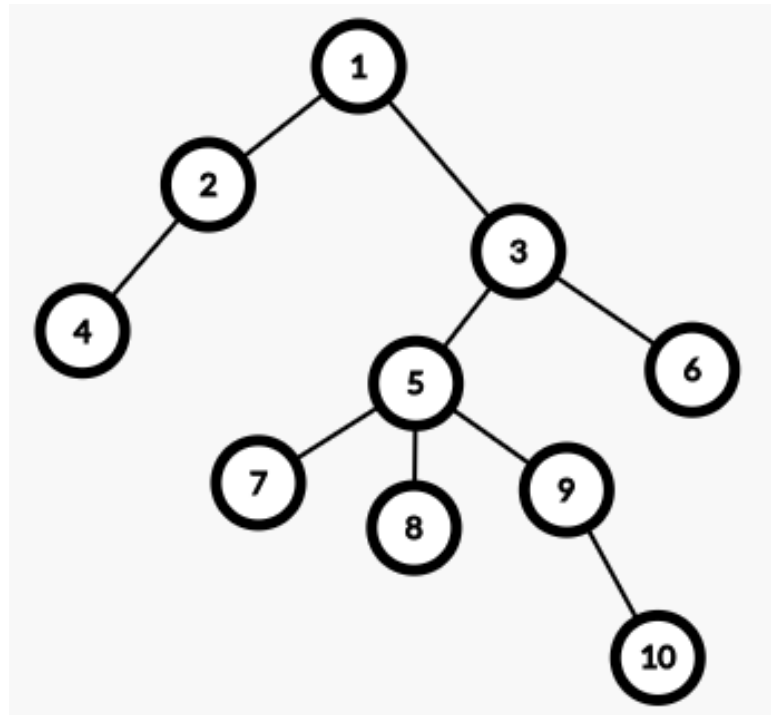
- Si es de tipo A, imprime un entero indicando la cantidad de generaciones descendientes que se salvarían.
- Si es de tipo B, imprime un entero indicando la cantidad total de personas que se salvarían, incluyendo a la propia persona.

## Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 10                   | 4                   |
| 1 2                  | 10                  |
| 3 1                  | 0                   |
| 4 2                  | 1                   |
| 6 3                  | 3                   |
| 3 5                  | 7                   |
| 5 7                  |                     |
| 8 5                  |                     |
| 9 5                  |                     |
| 10 9                 |                     |
| 6                    |                     |
| A 1                  |                     |
| B 1                  |                     |
| A 10                 |                     |
| B 10                 |                     |
| A 3                  |                     |
| B 3                  |                     |

## Nota

Para el primer caso de ejemplo, el árbol familiar tiene la estructura mostrada en la siguiente figura.



En las dos primeras consultas:

- La máxima generación descendiente salvada por la persona 1 es 4.
- El número total de personas salvadas es 10, incluyendo a la propia Iris.

En la quinta consulta, la persona 3 salva 3 generaciones descendientes y, en total, se salvan 7 personas.

# Problema E

## Día de la Amistad

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Marco Reus*

Para el Día de la Amistad, Reus decidió enviar postales a sus  $N$  amigos. Quiere hacer las postales con sus propias manos. Para ello, tiene una hoja de papel de tamaño  $X \times Y$ , que puede cortar en varias piezas.

Reus puede cortar cualquier hoja de papel de tamaño  $X \times Y$  únicamente en los siguientes casos:

- Si  $X$  es par, puede cortar la hoja por la mitad y obtener dos hojas de tamaño  $\frac{X}{2} \times Y$ .
- Si  $Y$  es par, puede cortar la hoja por la mitad y obtener dos hojas de tamaño  $X \times \frac{Y}{2}$ .
- Si  $X$  y  $Y$  son pares al mismo tiempo, puede elegir cualquiera de las dos formas anteriores para realizar el corte.

Cada vez que corta una hoja de papel, el número total de hojas aumenta en 1.

Ayuda a Reus a determinar si puede cortar su hoja de tamaño  $X \times Y$  en al menos  $N$  piezas, utilizando únicamente las reglas descritas anteriormente.

### Entrada

La primera línea contiene un entero  $t$  ( $1 \leq t \leq 10^4$ ), que representa el número de casos de prueba.

Cada caso de prueba consiste en una línea con tres enteros  $X$ ,  $Y$  y  $N$  ( $1 \leq X, Y \leq 10^4$ ,  $1 \leq N \leq 10^9$ ), donde:

- $X$  es el ancho de la hoja.
- $Y$  es la altura de la hoja.
- $N$  es el número de amigos a quienes Reus debe enviar una postal.

### Salida

Para cada caso de prueba, imprime una línea con:

- SI si es posible cortar la hoja de tamaño  $X \times Y$  en al menos  $N$  piezas.
- NO en caso contrario.

## Ejemplos

| Ejemplo de entrada 1                                    | Ejemplo de salida 1        |
|---------------------------------------------------------|----------------------------|
| 5<br>12 3 4<br>3 12 5<br>1 1 10<br>24 18 14<br>24 18 20 | SI<br>NO<br>NO<br>SI<br>NO |

# Problema F

## F maestro

*Tiempo Límite: 1 seg, Memoria Límite: 128 MB*

*Autor: Romer Flores Condori*

Domer se encuentra en el viaje a la SansiCup2025 pero olvido terminar sus pendientes, entre ellos olvido que tenia el examen mas importante de su vida. Al no poder realizarlo por que el bus lo dejo en la carretera todo de nieve te llamo a ti para resolverlo.

El enunciado dice: “Dado un numero  $N$ , hallar tres numeros enteros positivos tal que sumados resulte  $N$ ”

### Entrada

Un solo numero entero  $N$  ( $3 \leq n \leq 10^5$ )

### Salida

Imprime tres numeros enteros separados por un espacio  $a > 0, b > 0$  y  $c > 0$  tal que  $a + b + c = N$ , si hay multiples respuestas validas puedes imprimir cualquiera de ellas.

### Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 5                    | 2 1 2               |

| Ejemplo de entrada 2 | Ejemplo de salida 2 |
|----------------------|---------------------|
| 30                   | 14 3 13             |

# Problema G

## Jean Grey

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*  
*Autor: Romer Flores Condori*

Tras los acontecimientos de *Spider-Man: Brand New Day*, el Departamento de Control de Daños (DODC) desarrolló un componente experimental con el objetivo de debilitar las habilidades telepáticas de Jean Grey.

Antes del experimento, Jean podía transferir instantáneamente su consciencia al cerebro de cualquier persona ubicada dentro de un radio de 9 metros, tomando el control absoluto de su cuerpo mientras el anfitrión quedaba inconsciente. Además, podía repetir este proceso de una persona a otra en cuestión de instantes, dando la impresión de teletransportarse entre la multitud.



Sin embargo, el componente alteró el funcionamiento de su poder. Ahora, la posibilidad de poseer a una persona ya no depende únicamente de la distancia, sino también del **primer nombre** de la víctima.

Antes de intentar la posesión, Jean puede reorganizar las letras del nombre en cualquier orden.

Jean podrá poseer a una persona si es posible reordenar su nombre de manera que se cumpla la siguiente condición:

- Para todo par de posiciones  $i < j$  tales que  $S_i = S_j$ , la distancia entre ellas ( $j - i$ ) debe ser impar.

En otras palabras, dos letras iguales nunca pueden encontrarse a una distancia par entre sí.

Dado el primer nombre de una persona, determine si Jean Grey podrá poseerla.

### Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 10^2$ ), la longitud del nombre.

La segunda línea contiene una cadena  $S$  de longitud  $N$ , formada únicamente por letras minúsculas del alfabeto inglés.

## Salida

Imprima SI si es posible reordenar el nombre para que Jean pueda poseer a la persona.

En caso contrario, imprima NO.

## Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 8<br>abcdabcd        | SI                  |

| Ejemplo de entrada 2 | Ejemplo de salida 2 |
|----------------------|---------------------|
| 8<br>aabccbd         | NO                  |

| Ejemplo de entrada 3 | Ejemplo de salida 3 |
|----------------------|---------------------|
| 7<br>aabccbd         | SI                  |

# Problema H

## La promesa del Capitán Cornelio

*Tiempo Límite: 2 seg, Memoria Límite: 256 MB*

*Autor: Miguel Angel Quispe Mamani*

En el **País de las Maravillas**, el legendario **Capitán Cornelio** protagoniza diariamente divertidas conferencias donde comenta todo lo que sucede en el reino.



Un día, después de una acalorada discusión con el gobernante del reino, conocido únicamente como **Don Bigotes**, el Capitán Cornelio hizo una solemne promesa frente a todos los habitantes:

“A partir de hoy no te voy a nombrar”.

Sin embargo, como todos en el País de las Maravillas conocen el peculiar carácter del Capitán Cornelio, aquella promesa duró muy poco. En sus siguientes apariciones volvió a mencionar a Don Bigotes una y otra vez.

Intrigados por este comportamiento, los sabios de la **Instancia de Control y Prueba de Contenidos (ICPC)** comenzaron a registrar cuántas veces el Capitán Cornelio mencionaba a Don Bigotes cada día.

Durante  $N$  días consecutivos, la ICPC almacena esta información en un vector de enteros  $A$ , donde  $A_i$  representa la cantidad de veces que el Capitán Cornelio mencionó a Don Bigotes durante el  $i$ -ésimo día.

La ICPC considera que existe una **racha creciente** cuando, durante varios días consecutivos, el número de menciones realizadas cada día es estrictamente mayor que el número de menciones del día anterior.

Tu tarea es determinar la longitud máxima de una racha creciente de días consecutivos.

Por ejemplo, si se tiene el siguiente vector:

$$A = [4, 2, 5, 6, 8, 10, 6, 7, 8, 3]$$

la racha creciente más larga es:

[2, 5, 6, 8, 10]

Por lo tanto, la respuesta es 5.

## Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 10^6$ ), que representa la cantidad de días registrados.

La segunda línea contiene  $N$  números enteros  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^9$ ), donde  $A_i$  representa la cantidad de veces que el Capitán mencionó al gobernante durante el  $i$ -ésimo día.

## Salida

Imprime un único número entero que indique la máxima cantidad de días consecutivos durante los cuales el número de menciones fue estrictamente mayor que el del día anterior.

## Ejemplos

| Ejemplo de entrada 1       | Ejemplo de salida 1 |
|----------------------------|---------------------|
| 10<br>4 2 5 6 8 10 6 7 8 3 | 5                   |

| Ejemplo de entrada 2 | Ejemplo de salida 2 |
|----------------------|---------------------|
| 8<br>9 8 6 5 3 2 1 1 | 1                   |

## Nota

En el primer ejemplo, la racha creciente más larga está formada por:

[2, 5, 6, 8, 10]

Esta racha tiene longitud 5.

En el segundo ejemplo, no existen dos días consecutivos en los que el número de menciones aumente.

Por lo tanto, cada día forma individualmente una racha creciente de longitud 1, y la respuesta es 1.

# Problema I

## Los cuadernos aesthetic de Leif

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Luis David Ajuacho Tarquino*

Leif acaba de defender exitosamente su tesis. Después de la defensa, sus amigos le dijeron que su etapa universitaria había terminado y que, por fin, debía deshacerse de los materiales que acumuló durante todos esos años.

Sin embargo, Leif no quiere tirar sus cuadernos. Algunos contienen ejercicios incompletos, otros fórmulas que ya no recuerda y varios están llenos de anotaciones que probablemente nunca volverá a leer. Aun así, cada uno representa una parte importante de su vida universitaria.

Por esta razón, Leif decidió conservarlos y usarlos para crear una decoración *aesthetic* en su habitación.

Leif distribuyó todos sus cuadernos en  $n$  pilas alineadas horizontalmente y numeradas del 1 al  $n$ , de izquierda a derecha. Inicialmente, cada pila contiene exactamente  $k$  cuadernos.

Sea  $a_i$  la cantidad de cuadernos de la pila  $i$ . Leif considera que una disposición es *aesthetic* si cada pila contiene estrictamente más cuadernos que la pila ubicada inmediatamente a su izquierda. Es decir,

$$a_1 < a_2 < \dots < a_n.$$

Para intentar obtener una disposición *aesthetic*, Leif puede realizar la siguiente operación cualquier cantidad de veces, posiblemente ninguna:

1. Elegir dos pilas diferentes  $i$  y  $j$ . Las pilas pueden encontrarse en cualquier posición y no necesitan ser adyacentes.
2. Elegir un entero  $x$  tal que  $1 \leq x < a_i$ .
3. Retirar exactamente  $x$  cuadernos de la parte superior de la pila  $i$  y colocarlos sobre la pila  $j$ .

Una misma pila puede participar en varias operaciones, ya sea entregando o recibiendo cuadernos. Leif no puede cambiar el orden de las pilas, dejar una pila vacía, crear cuadernos nuevos ni destruir alguno de los que ya tiene.

Determina si Leif puede reorganizar sus cuadernos de manera que la disposición final sea *aesthetic*.

### Entrada

La primera línea contiene un entero  $t$  ( $1 \leq t \leq 10^3$ ), que representa la cantidad de casos de prueba.

Cada caso de prueba contiene dos enteros  $n$  y  $k$  ( $1 \leq n, k \leq 10^9$ ), donde:

- $n$  es la cantidad de pilas de cuadernos.
- $k$  es la cantidad inicial de cuadernos en cada pila.

## Salida

Para cada caso de prueba, imprime SI si Leif puede obtener una disposición *aesthetic*. En caso contrario, imprime NO.

## Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 6                    | SI                  |
| 5 4                  | NO                  |
| 4 2                  | SI                  |
| 4 3                  | SI                  |
| 1 7                  | NO                  |
| 6 3                  | SI                  |
| 6 4                  |                     |

## Nota

En el primer caso de prueba, Leif comienza con la disposición:

[4, 4, 4, 4, 4]

Después de realizar algunas operaciones, puede obtener:

[1, 2, 3, 4, 10].

Esta disposición es *aesthetic*.

En el segundo caso de prueba, no existe una forma válida de distribuir los cuadernos entre las cuatro pilas para que sus cantidades sean estrictamente crecientes.

En el cuarto caso de prueba, solamente existe una pila. Por lo tanto, la disposición ya cumple la condición.

# Problema J

## Suertudo

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*  
*Autor: Romer Flores Condori*

Nos volvemos a encontrar con Domer el ludopata, despues de ganar millones de chelines por apostar a que España ganaba La Copa Mundial de Futbol 2026, ahora quiere invertir todo ganado en la loteria.

Por lo cual Domer decide comprar  $t$  boletos de loteria, donde cada boleto consta de 6 digitos. Puedes cuestionar a Domer el ludopata pero no puedes cuestionar su suerte, ya que en la mañana del sorteo, el pasante de la Loteria comete un error en el sistema de del sorteo, por lo cual se declara ganador a aquel billete que cumpla que la suma de sus primeros 3 digitos sea igual a la suma de sus ultimos 3 digitos.

Como Domer tiene muchos boletos, no sabe como clasificarlos, asi que pidio tu ayuda para saber cuales son ganadores y cuales no.

### Entrada

La primera linea consta de un entero  $t$  ( $1 \leq t \leq 1000$  que representa la cantidad de boletos que compro Domer, luego siguen  $t$  lineas donde cada linea consta de 6 digitos.

### Salida

Por cada billete debes responder “SI”, si este es un boleto ganador y “NO” si este es un boleto perdedor.

### Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 5                    | SI                  |
| 002101               | NO                  |
| 123456               | NO                  |
| 314159               | SI                  |
| 333333               | SI                  |
| 050131               |                     |

# Problema K

## Bron

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Los maquinolas: The last dance*

Dylan, un intrépido boliviano, llegó a Buenos Aires decidido a descubrir el verdadero secreto de las empanadas argentinas. Su plan era simple: recorrer todos los puestos que encontrara y comprar una empanada en cada uno.

Cada puesto tiene su propio precio, y Dylan anotó cuidadosamente cuánto le costó cada empanada. Sin embargo, un misterioso vendedor le advirtió:

“Si comes demasiadas empanadas del mismo precio... ¡vas a explotar, Kbron!”

Como Dylan quiere seguir vivo para probar el dulce de leche, necesita saber cuál es la mayor cantidad de empanadas que tienen exactamente el mismo precio.

Tu tarea es ayudar a Dylan antes de que sea demasiado tarde.



### Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 1000$ ), la cantidad de empanadas que Dylan compró.

La segunda línea contiene  $N$  enteros  $a_1, a_2, \dots, a_N$ , donde  $0 \leq a_i \leq 10000$ , representando el precio de cada empanada.

### Salida

Imprime un único entero: la máxima cantidad de veces que aparece un mismo precio entre todas las empanadas.

## Ejemplos

| Ejemplo de entrada 1      | Ejemplo de salida 1 |
|---------------------------|---------------------|
| 7<br>10 20 10 15 20 10 30 | 3                   |

| Ejemplo de entrada 2         | Ejemplo de salida 2 |
|------------------------------|---------------------|
| 8<br>50 40 30 20 10 60 70 80 | 1                   |

| Ejemplo de entrada 3                | Ejemplo de salida 3 |
|-------------------------------------|---------------------|
| 10<br>25 25 25 30 30 40 40 40 40 25 | 5                   |

# Problema L

## Un Jardín Armonioso

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Kelvin Bohn Davisson Vargas Garcia*

Eres el guardián de un bosque mágico y te han entregado una fila  $a$  que consta de  $n$  flores resplandecientes. Cada flor tiene un nivel de luminosidad representado por un número entero. Tu tarea es encontrar la longitud del prefijo más pequeño (el más corto) de flores que debes podar (eliminar) desde el principio de la fila  $a$  para que la fila restante sea un jardín *armonioso*.

Recuerda que un prefijo de la fila  $a = [a_1, a_2, \dots, a_n]$  es una subfila que contiene las primeras flores: el prefijo de la fila  $a$  de longitud  $k$  es el arreglo  $[a_1, a_2, \dots, a_k]$  ( $0 \leq k \leq n$ ).

Una fila de flores  $b$  de longitud  $m$  se considera *armoniosa* si puedes obtener a partir de ella un nuevo arreglo  $c$  con luminosidades en orden **no decreciente** ( $c_1 \leq c_2 \leq \dots \leq c_m$ ) repitiendo la siguiente operación  $m$  veces (inicialmente,  $c$  está vacío):

- Selecciona la primera flor o la última flor de la fila  $b$ , retírala de  $b$ , y plántala al final del arreglo  $c$ .

Por ejemplo, si hacemos 4 operaciones: tomamos  $b_1$ , luego  $b_m$ , luego  $b_{m-1}$  y por último  $b_2$ , entonces  $b$  se convierte en  $[b_3, b_4, \dots, b_{m-3}]$  y  $c = [b_1, b_m, b_{m-1}, b_2]$ .

Considera la siguiente fila de flores con luminosidades  $b = [1, 2, 3, 4, 4, 2, 1]$ . Esta fila es armoniosa porque podemos obtener un arreglo no decreciente  $c$  mediante la siguiente secuencia de operaciones:

1. Tomamos la primera flor de  $b$ , por lo que  $b = [2, 3, 4, 4, 2, 1]$  y  $c = [1]$ .
2. Tomamos la última flor de  $b$ , por lo que  $b = [2, 3, 4, 4, 2]$  y  $c = [1, 1]$ .
3. Tomamos la última flor de  $b$ , por lo que  $b = [2, 3, 4, 4]$  y  $c = [1, 1, 2]$ .
4. Tomamos la primera flor de  $b$ , por lo que  $b = [3, 4, 4]$  y  $c = [1, 1, 2, 2]$ .
5. Tomamos la primera flor de  $b$ , por lo que  $b = [4, 4]$  y  $c = [1, 1, 2, 2, 3]$ .
6. Tomamos la última flor de  $b$ , por lo que  $b = [4]$  y  $c = [1, 1, 2, 2, 3, 4]$ .
7. Tomamos la única flor restante en  $b$ , por lo que  $b = []$  y  $c = [1, 1, 2, 2, 3, 4, 4]$ . El arreglo  $c$  es no decreciente, por lo que es un jardín perfecto.

Ten en cuenta que una fila compuesta por una sola flor siempre se considera armoniosa.

Imprime la longitud del prefijo más corto de  $a$  que debes podar para que  $a$  se convierta en una fila armoniosa. Ten en cuenta que la longitud requerida puede ser 0.

Debes responder  $t$  casos de prueba independientes.

## Entrada

La primera línea contiene un único número entero  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — la cantidad de flores en la fila  $a$ . La segunda línea del caso de prueba contiene  $n$  números enteros  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 2 \cdot 10^5$ ), donde  $a_i$  es el nivel de luminosidad de la  $i$ -ésima flor de la fila.

## Salida

Imprime la respuesta en una nueva línea: la longitud del prefijo más corto de flores que debes podar de  $a$  para hacer que la fila sea armoniosa.

## Ejemplos

| Ejemplo de entrada 1 | Ejemplo de salida 1 |
|----------------------|---------------------|
| 5<br>5 4 3 2 3       | 3                   |

| Ejemplo de entrada 2 | Ejemplo de salida 2 |
|----------------------|---------------------|
| 7<br>1 3 1 4 5 3 2   | 2                   |

| Ejemplo de entrada 3 | Ejemplo de salida 3 |
|----------------------|---------------------|
| 3<br>1 1 1           | 0                   |

# Problema M

## El robo final

*Tiempo Límite: 1 seg, Memoria Límite: 256 MB*

*Autor: Eliseo Eliezer Condori Mamani*

Marco es un estudiante de informática que guarda sus  $t$  cosas más valiosas en diferentes casilleros en la Facultad de Informática. Están asegurados por un candado que tiene un funcionamiento interesante: el candado tiene solo cuatro combinaciones y funciona como un círculo en el plano cartesiano.

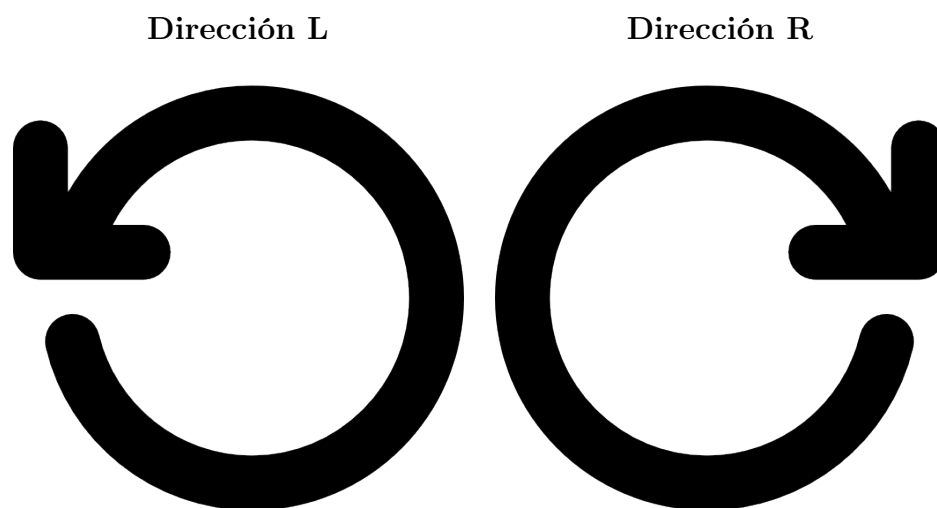
El candado está inicialmente en uno de estos puntos:

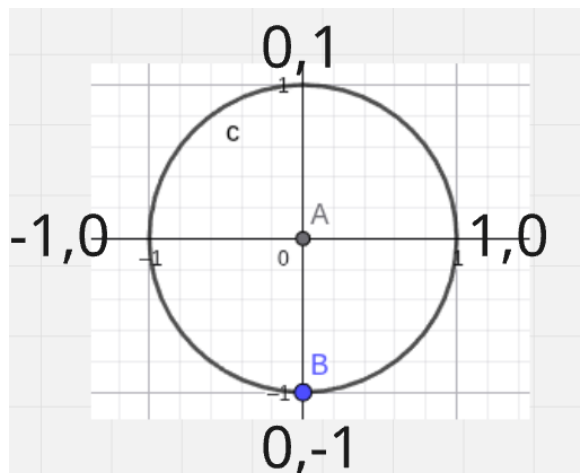
$$(x, 0), \quad (0, x), \quad (-x, 0), \quad (0, -x)$$

y para desbloquearlo tienes que descubrir en cuál de esas posiciones está y presionar un botón. Si encuentras la combinación correcta, ganarás el premio del casillero (quizás solo una pelota de baloncesto), pero si fallas, se activará un terrible sonido y tendrás que correr.

Marco perdió las instrucciones para desbloquear los casilleros, y tú las encontraste, pero te das cuenta de que no escribió la posición correcta; escribió  $m$  instrucciones de giro, con la dirección del giro y la cantidad de pasos a seguir. No sería un problema, pero Marco escribió números muy grandes. ¿Serás capaz de determinar la respuesta correcta?

Si la dirección es derecha (**R**), los pasos avanzan en sentido horario; si la dirección es izquierda (**L**), avanzan en sentido antihorario.





*Para cada candado existen únicamente cuatro posiciones posibles.*

## Entrada

- La primera línea contiene dos enteros  $a$  y  $b$ , donde exactamente uno de ellos es cero (como se describió anteriormente) y el otro pertenece al rango

$$[-10^9, -1] \cup [1, 10^9],$$

indicando la posición inicial del candado.

- La siguiente línea contiene un entero  $m$ , el número de instrucciones escritas por Marco.
- Cada una de las siguientes  $m$  líneas contiene un carácter  $x$  ( $x \in \{L, R\}$ ) y un entero  $y$ , que representan la dirección del giro y la cantidad de pasos, respectivamente.

## Salida

Imprime dos enteros que representen la combinación correcta que desbloquea el candado.

## Ejemplos

| Ejemplo de entrada 1    | Ejemplo de salida 1 |
|-------------------------|---------------------|
| -1 0<br>2<br>R 1<br>L 1 | -1 0                |

| Ejemplo de entrada 2   | Ejemplo de salida 2 |
|------------------------|---------------------|
| 3 0<br>2<br>R 2<br>L 3 | 0 3                 |

| Ejemplo de entrada 3           | Ejemplo de salida 3 |
|--------------------------------|---------------------|
| 0 -2<br>3<br>R 3<br>L 2<br>R 3 | 0 -2                |