Concurso de Programación

Concurso I 2025 - Estructura de Datos

https://urjc-cp.github.io/urjc-cp/

Cuadernillo de problemas

Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
21 de febrero de 2025

Índice

Autores de los problemas:

- Iván Penedo Ventosa (Universidad Rey Juan Carlos)
- Adaya Ruiz Mayoral (GMV, Universidad Rey Juan Carlos)
- Lucas Martín García (Universidad Rey Juan Carlos)
- Sara García Rodríguez (Universidad Rey Juan Carlos)
- Alicia Pina Zapata (Universidad Rey Juan Carlos)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Sergio Salazar Cardenas (Universidad Rey Juan Carlos)

Testers no autores de los problemas:

- Raúl Fauste Jimenez (Universidad Rey Juan Carlos)
- David Enrique Orna Alcobendas (Universidad Rey Juan Carlos)
- Alejandro Mayoral Gómez (Universidad Rey Juan Carlos)
- Eva Gómez Fernández (Universidad Rey Juan Carlos)

Tiempo: 1 segundo

A
RomanoX

Elmo, conocido por su carisma y su habilidad para conectar con la gente, hace poco tiempo asistió a un mitin muy importante. Durante su discurso, en un momento de inspiración, levantó el brazo y realizó el saludo romano. Este gesto, combinado con su número favorito, el 10, dejó una impresión duradera en todos los presentes. Desde ese día, Elmo fue apodado "RomanoX" por sus seguidores y amigos.

La fama de Elmo creció rápidamente, y con ella, la peculiar costumbre de sus seguidores de comunicarse con él utilizando números romanos. Cada vez que alguien le escribía una carta, un mensaje o incluso una nota rápida, usaban números romanos para expresar cualquier cifra. Aunque al principio Elmo encontraba esto divertido, pronto se dio cuenta de que le tomaba mucho tiempo traducir los números romanos a números arábigos para poder entender los mensajes. Sobre todo, teniendo en cuenta que cuando un carácter aparece a la izquierda de otro con mayor valor, este primero resta su cantidad al carácter superior, algo que nunca ha entendido porqué.

Un día, abrumado por la cantidad de mensajes que recibía y la dificultad de traducirlos, Elmo recordó tú eras un experto en programación, así que ahora te ha pedido ayuda para diseñar un programa que pudiera traducir los números romanos a números arábigos de manera rápida y eficiente. Así, Elmo podría dedicar más tiempo a sus seguidores y menos tiempo a descifrar números.

Entrada

La entrada corresponde a una cadena de caracteres de longitud N que representa un número romano. La cadena puede contener los siguientes caracteres, con sus correspondientes valores: "I"= 1, "V"= 5, "X"= 10, "L"= 50, "C"= 100, "D"= 500 y "M"= 1000.

Salida

La salida será un único número entero X que indique el valor del número romano.

Entrada de ejemplo

```
X
```

Salida de ejemplo

```
10
```

Entrada de ejemplo

```
MMXXV
```

Salida de ejemplo

```
2025
```

Límites

- $1 \leq N \leq 20$.
- El número romano dado siempre será válido y estará en mayúsculas.

Tiempo: 1 segundo

B
Pillando tramposos

En la Universidad Rey Juan Carlos, la asociación de alumnos Dijkstraídos en colaboración al grupo de investigación GRAFO están impartiendo un curso de programación competitiva, para la cual es necesario una gran capacidad de resolución de problemas y un amplio conocimiento de algoritmos eficientes. Este curso no solo busca preparar a los estudiantes para competiciones nacionales, sino que también ofrece la oportunidad de representar a la universidad en prestigiosos concursos internacionales por toda Europa. Estos viajes no solo son una oportunidad para demostrar sus habilidades, sino también para conocer nuevas culturas y establecer contactos con otros apasionados de la programación.

Para muchos de los participantes, este curso representa su último año en la universidad, y por ello, se está poniendo un gran énfasis en formar a las nuevas promesas de la programación competitiva. Sin embargo, en un intento por asegurarse una plaza en el concurso internacional, algunos estudiantes han comenzado a utilizar inteligencias artificiales generativas, como Chat GePeTo, para resolver los problemas de programación de manera más rápida y eficiente.

Cansados de que algunos estudiantes se aprovechen de estas herramientas, los organizadores del curso han decidido implementar un nuevo mecanismo para detectar cuándo un alumno está utilizando inteligencia artificial en algunos problemas o si directamente la está usando para todos los problemas con el fin de resolverlos lo más rápido posible. Para ello, quieren desarrollar un programa que, dados los tiempos estimados en resolver cada uno de los problemas de varios concursos y los tiempos que tardan cada uno de los alumnos, determine cuáles de ellos saben de programación y están resolviendo los problemas por sí mismos, cuáles están usando IA en algunos problemas y cuáles están abusando de la IA y usando herramientas como Chat GePeTo para todos los problemas.

Con esto no solo buscan mantener la integridad de la competición, sino también asegurar que los estudiantes realmente desarrollen las habilidades necesarias para sobresalir en el campo de la programación y que sean merecedores del reconocimiento que dan los concursos tanto nacionales como internacionales. ¿Podrías echarles una mano?

Entrada

La entrada de este problema estará formada por varios casos de prueba donde cada caso de prueba es un concurso de programación competitiva, hasta final de fichero.

Para cada caso de prueba la primera línea estará formada por dos enteros N y M separados por un espacio, donde N es el número de problemas del concurso y M es el número de alumnos que han participado. Tras este, se darán N números en una única línea que indican los minutos n_i que se estima que tarden los alumnos en resolver el problema i del concurso.

En cada una de las siguientes M líneas se darán N números en cada línea donde se representan los minutos m_{ij} que tarda en realizar el problema i el alumno j.

Salida

Para cada caso de prueba, deberás devolver M líneas donde en cada una se indicará si un alumno usa IA total o parcialmente, o si por el contrario no la usa. Para esto deberá imprimir:

- "Usa IA" si estamos seguros de que usa IA.
- "Sospechoso" si no estamos seguros pero lo parece.
- "Le sabe" si nos fiamos de él.

Al final de cada caso de prueba, tendrás que imprimir "-----" para dividir los concursos.

Entrada de ejemplo

```
7 3
2 10 6 15 30 47 26
1 8 3 12 23 11 24
5 20 19 32 54 53 26
20 10 13 16 24 39 37
```

Salida de ejemplo

```
Usa IA
Le sabe
Sospechoso
-----
```

Entrada de ejemplo

```
2 1
16 24
15 24
3 2
4 2 3
7 4 7
1 1 2
```

Salida de ejemplo

```
Sospechoso
-----
Le sabe
Usa IA
-----
```

Límites

- $1 \leq N, M \leq 10000$
- $1 \leq n_i, m_{ij} \leq 100000$

Tiempo: 1 segundo

C
Codazos en el metro

Subirse al metro a las nueve de la mañana es una tarea complicada. No por madrugar, que también, si no porque está hasta arriba de gente. A veces la gente tiene que apelotonarse para poder subir y luego como no estés cerca de la puerta cuando te toque bajar tienes un problema.

La única manera de que no haya problemas es que a lo largo de todas las paradas la gente se baje en el orden contrario al que ha subido. Es decir, que el último en subir sea siempre el primero en bajar.

Esta mañana he estado anotando a lo largo de unas cuantas paradas la gente que se subía y bajaba del metro y en qué orden. ¿Sabrías decirme si ha habido algún problema? Hay un problema cuando alguien que se baja no estaba ya en la puerta, por lo que ha tenido que salir a codazos.

Cuando empiezo a anotar el metro está vacío y cuando termino no me importa si queda gente.

Entrada

La entrada corresponde a varios casos de prueba. La primera línea es un entero T que indica el número de casos de prueba. Cada caso de prueba empieza con un número N indicando cuántas paradas he anotado. Le siguen $2N$ líneas en las que se indican en una línea en orden las personas que se han bajado en esa parada y en la siguiente línea las personas que se han subido, también en orden. Cada persona está identificada por un número p único. Si en una línea hay un único número 0 significa que nadie ha bajado/subido en esa parada.

Salida

Para cada caso de prueba, la salida debe ser una línea en la que ponga "CODAZO" si ha habido algún problema en esas paradas, o "BIEN" si todo ha transcurrido sin problemas.

Entrada de ejemplo

```
2
3
0
2 3 5
5
4 10
10 4 3
11 8
2
0
1 7 6 4
6 4
0
```

Salida de ejemplo

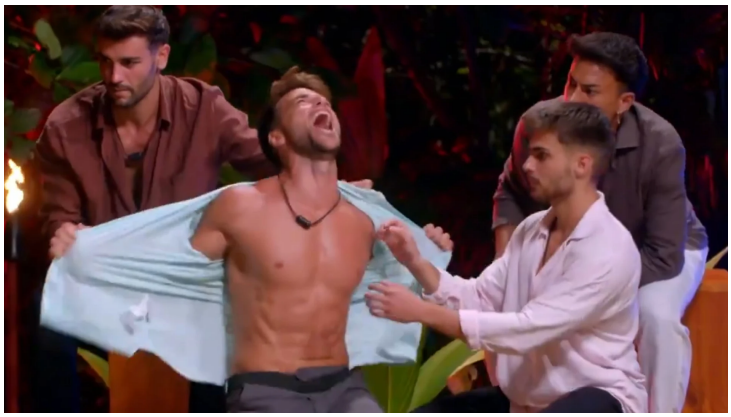
```
BIEN
CODAZO
```

Límites

- $1 \leq p \leq 10^9$
- A lo largo de todos los casos el número de personas que se suben y bajan es menor que 10^6

Tiempo: 2 segundos

D
Montayo por favor!!



Nuestro amigo Montayo ha decidido participar en un conocido reality llamado “La isla de las conexiones” con su novia Anati para ver si realmente están preparados para el compromiso. Montayo ya lleva unos días un poco mosca por el comportamiento de su novia, que se está acercando demasiado a un chico llamado Manel. Después de ver imágenes de ambos teniendo una conexión demasiado real, Montayo ha llegado a su límite y ha decidido romper las normas para ir a buscar a Anati y pedirle explicaciones.

El problema de Montayo es que, después de correr sin rumbo por la playa, no ha conseguido averiguar dónde está la villa en la que está su novia, así que necesita tu ayuda para encontrarla. Las villas de esta isla tienen una arquitectura muy peculiar, hay N villas de alturas diferentes (no hay dos villas que tengan la misma) situadas en línea recta, y todas están a un kilómetro de distancia. Montayo sabe que la villa de Anati es más alta que su villa, y que es la más próxima que cumple esa condición. Por lo tanto, necesitamos calcular la mínima distancia de cada villa a la más próxima que sea más alta que ella, ya que Montayo no sabe a cuál corresponde cada altura – lo averiguará cuando consiga robar un mapa de la isla a Sendra, la presentadora, y no tendrá tiempo para hacer cálculos antes de salir corriendo.

Entrada

La entrada corresponde a un único caso de prueba. La primera línea es un entero N, que corresponde al número de villas. La segunda línea contiene N números correspondientes a la altura H_i de cada una de las villas, siendo i el índice de cada una (H_0 es la altura de la villa más a la izquierda, H_1 la altura de la villa a la derecha de H_0, etc.).

Salida

La salida debe contener la distancia $d(i)$ de cada villa a la villa más cercana con altura superior a la suya:

$$d(i) = \text{mín}\{|i - j| \text{ para cada } j \text{ tal que } H_j > H_i\}$$

En caso de que no haya ninguna villa que cumpla esa condición, $d(i)$ deberá ser 0.

Entrada de ejemplo

```
5
7 3 2 100 1
```

Salida de ejemplo

```
3 1 1 0 1
```

Entrada de ejemplo

```
8
45 13 18 10 8 56 17 19
```

Salida de ejemplo

```
5 1 2 1 1 0 1 2
```

Límites

- $2 \leq N \leq 200000$
- $0 \leq H_i \leq 10^{18}$, para $i = 0, ..., N-1$

Tiempo: 1 segundo

E
Cuántas complejidades

Los profesores del curso de programación competitiva se fueron de cervezas en vez de preparar el concurso de estructuras de datos, por lo que no han podido preparar el juez automático. Como consecuencia de sus actos, ahora les toca evaluar a mano todos los envíos que hagan los alumnos durante el concurso.

Para ello, tienen que ejecutar el código que entreguen los alumnos y un programa que han preparado en 5 segundos con ChatGPT les devuelve el número de instrucciones por segundo que ejecutan. Con esta información, y con ayuda de la tabla de complejidades que explicaron en la última clase, deben ser capaces de saber qué complejidad tiene el código que los alumnos han enviado.

Complejidad	Operaciones por segundo
`O(1)`	:D
`O(n)`	$1,000,000$
`O(n*logn)`	$100,000$
`O(n^2)`	1000
`O(n^3)`	100
`O(2^n)`	20
`O(n!)`	12

Como en verdad nadie tiene ni idea de programar, parece que te va a tocar a ti echarles una mano. ¿Podrías ayudarles?

Entrada

En la primera línea se dará un número entero T que representa el número de casos de prueba, seguido de T líneas, cada una con un número entero T_i entre 1 y 10^9, que representa el número de instrucciones por segundo que ejecuta el código de un alumno.

Salida

Para cada código ejecutado, imprime la complejidad correspondiente según la tabla dada.

Entrada de ejemplo

```
3
19
20
21
```

Salida de ejemplo

```
O(2^n)
O(2^n)
O(n^3)
```

Límites

- $1 \leq N \leq 10^9$
- $1 \leq T_i \leq 10^9$
- CUIDADO! No hay que imprimir espacios en ninguna complejidad!

Tiempo: 1 segundo

F
Los problemas de la tecnología

Todos los 22 de diciembre en España es tradición el sorteo extraordinario de Navidad. Ya son más de 200 años desde 1812 que se realiza, este sorteo reparte un 70 % de la emisión, que este 2022 fue de 2.520 millones de euros, siendo el mayor premio de 400.000 euros al décimo.

El sorteo comienza sobre las 9 de la mañana, lo primero que realizan es la inserción de todas las bolas con todos los números del 0 al 99999 junto a las 1808 bolas con premio. El procedimiento del sorteo consiste en extraer una bola con un premio, nombrarlo en voz alta y así hasta finalizar el sorteo.

Las páginas webs saben que este día mucha gente está con mucha prisa de conocer si ha obtenido un premio con sus décimos de lotería. Para ser los primeros, lo que hacen muchas de ellas es usar un sistema que traduce el número nombrado junto al premio y lo añade en su base de datos.

Todos los humanos nos equivocamos, en 2021, 54 números fueron cantados erróneamente llamados los números fantasma. Los organizadores del evento dicen reiteradas veces que no se fíen de otras webs que no sean la oficial y esperen al día siguiente para verificar los números. El fiarse de otras webs que no sean la oficial, provocan que millones de euros no sean cobrados por no ir a comprobar el décimo.

Los organizadores quieren montar un sistema que verifique los diferentes números para cualquier sorteo. Dónde dado un listado con los números oficiales y de cualquier web los números que han publicado.

Entrada

La entrada consistirá en un único caso de prueba, donde la primera línea es un número P con el total de números premiados. Las dos líneas siguientes tendrán P números, siendo la primera línea un listado con P_i números con los números oficiales del sorteo y la segunda línea con los números no oficiales. El sistema asegura que en cada línea no existirán números repetidos.

Salida

La salida de prueba será "OK" en el caso de que todos los números del listado no oficial coincidan con el listado oficial. Por el contrario, si hubiese un número o más del listado oficial que no apareciese en el listado no oficial, estos aparecerán ordenados de forma creciente.

Entrada de ejemplo

```
3
1 2 3
1 3 2
```

Salida de ejemplo

```
OK
```

Entrada de ejemplo

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

Salida de ejemplo

```
3
```

Límites

- $1 \leq P \leq 10^5$ y $1 \leq P_i \leq 2^{63} - 1$

Tiempo: 1 segundo

G
Menú Amigo

Es viernes y el kebab de tu barrio ha sacado por fin su famoso Menú Amigo en colaboración con tu cantante favorito. Como sabían que se iba a liar, han decidido utilizar un nuevo sistema con el que esperan evitar peleas: cada persona que llega coge un ticket con número (los números siguen un orden ascendente) y se pone a la cola.

Cuando has querido llegar, después de salir del pesado curso de programación competitiva, te has encontrado con una cola que daba la vuelta a la esquina, y no te ha quedado más remedio que coger tu número y ponerte a esperar.

Sin embargo, no has podido evitar darte cuenta de que hay gente que se está aprovechando de tener amigos en la cola para saltarse la espera: cogen su ticket pero en lugar de ponerse al final, se unen a su grupo de amigos.

Como te fastidia esperarte toda la cola y que no queden menús para cuando llegues, has decidido ir cotilleando los números que tiene la gente de la fila, para contar cuántos de ellos se han colado.

Dada la lista con los números de ticket que tiene cada persona en el orden de fila, ¿sabrías decir cuántos de ellos se han colado?

Entrada

La entrada comieinza con el número N de personas que hay en la fila.

En la siguiente línea se indican los N números de tickets, por orden de espera.

Salida

La salida se corresponderá con el número de personas que se han colado.

Entrada de ejemplo

```
6
1 3 12 5 8 9
```

Salida de ejemplo

```
1
```

Entrada de ejemplo

```
10
6 10 11 7 8 9 15 20 17 22
```

Salida de ejemplo

```
3
```

Límites

- $1 \leq N \leq 1000000$