



Concurso de Programación Concurso II 2026 - ED y Grafos

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

Cuadernillo de problemas



Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
6 de marzo de 2026

Índice

A Urgencias	3
B Óscar y los trenes	4
C La lista de la compra	6
D CTF	7
E Enseñando a sumar	9
F La ladrona de libros	10

Autores de los problemas:

- Iván Penedo Ventosa (Habla Computing, Universidad Rey Juan Carlos)
- Adaya Ruiz Mayoral (GMV, Universidad Rey Juan Carlos)
- Raúl Fauste Jiménez (Universidad Rey Juan Carlos)
- Sara García Rodríguez (Universidad Rey Juan Carlos)
- Sergio Salazar Cárdenas (Universidad Rey Juan Carlos)

Tiempo: 1 segundo

A Urgencias

Las urgencias del Hospital Severa Ocho están saturadas. El personal médico no da abasto con la cantidad de pacientes que necesitan ser atendidos. Para hacerlo lo mejor posible han decidido atender a los pacientes según el nivel de malestar que presenten.

En las urgencias del hospital hay 2 tipos de eventos:

- Puede llegar un nuevo PACIENTE. A este paciente se le hacen unas preguntas y se cuantifica su nivel de malestar.
- Pueden quedarse libres algunos médicos y ATENDER a pacientes que estén esperando. Se les atiende por nivel de malestar.

Entrada

La entrada corresponde a un único caso de prueba. La primera línea es un entero E , que indica el número de eventos que suceden en el hospital. Le siguen E líneas con un string p_i y un número entero n_i separados por un espacio. El string p_i puede ser “PACIENTE”, si llega un nuevo paciente a urgencias, seguido del número n_i , indicando su nivel de malestar. La otra opción es que el string p_i sea “ATENDER”, seguido de un número n_i , indicando que se atiende a n_i pacientes.

Salida

La salida consistirá en una línea por cada paciente atendido indicando cuál era su nivel de malestar.

Entrada de ejemplo

```
6
PACIENTE 3
PACIENTE 10
ATENDER 1
PACIENTE 2
PACIENTE 3
ATENDER 3
```

Salida de ejemplo

```
10
3
3
2
```

Límites

- $1 \leq E \leq 10^6$
- $p_i \in \{\text{PACIENTE}, \text{ATENDER}\}$ con $i = 1, \dots, E$
- $1 \leq n_i \leq 10^9$ con $i = 1, \dots, E$

Tiempo: 0.5 segundo(s)



Óscar y los trenes

Este comienzo de año no ha sido el idílico para el Ministerio de Transportes. Los temporales sufridos en toda la península han producido destrozos en la red ferroviaria. RENFE, una vez pasado el temporal, se ha puesto manos a la obra y ha decidido comprobar la magnitud de los destrozos.

La red ferroviaria española es muy importante que sea una red conexas, de forma que una persona que quiera viajar de Valencia a Pamplona, pueda hacerlo únicamente utilizando como medio de transporte el tren. El Ministerio de Transportes se está atragantando con la tarea y necesita tu ayuda.

Entrada

La entrada corresponde a un único caso de prueba. La primera línea se compone de dos números enteros, N , las ciudades con estación de tren en España y M el número de conexiones restantes tras el temporal. Las siguientes M líneas, se componen de dos números, S y T representando que existe una conexión entre la ciudad S y la ciudad T . Las conexiones entre dos ciudades son bidireccionales, es decir, si existe una conexión entre S y T , significa que se puede usar para ir de S a T y viceversa.

Salida

La salida consistirá en un único número. Se deberá imprimir el mínimo número de conexiones necesarias para que la red ferroviaria española vuelva a ser conexas.

Entrada de ejemplo

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

Salida de ejemplo

```
2
```

Entrada de ejemplo

```
2 1
1 2
```

Salida de ejemplo

```
0
```

Límites

- $1 \leq N \leq 10^5$
- $0 \leq M \leq \frac{N(N-1)}{2}$
- $1 \leq S, T \leq 10^5$ con $S \neq T$

Tiempo: 0.1 segundo(s)



La lista de la compra

David y Eva han cumplido por fin el objetivo de irse a vivir juntos, pero se han dado cuenta de que compartir piso no es solo elegir el color de las cortinas. Uno de los puntos donde más chocan es en algo tan básico como hacer la compra semanal. David es una persona extremadamente metódica; le gusta planificar las cenas de toda la semana y anotar en su móvil los ingredientes exactos, hasta el último bote de especias. Eva, en cambio, es más de improvisar, pero siempre se asegura de que los básicos de la despensa, como la leche o el papel higiénico, estén bien cubiertos para que nunca falte de nada.

El problema es que sus estilos de compra no terminan de encajar cuando están en el supermercado. David lleva su lista digital con cantidades muy específicas, mientras que Eva va añadiendo productos al carrito según recuerda lo que falta en casa. Al final, acaban con una lista de la compra bastante caótica que mezcla cosas como “3 botes de mermelada de higo” con “24 rollos de papel higiénico” o “5 kilos de patatas”.

El momento crítico llega siempre en la línea de cajas. Entre el ruido del supermercado, la gente que empuja con las prisas y la velocidad a la que el cajero pasa los productos por el escáner, tus amigos suelen entrar en un estado de confusión total. Eva mira el carrito y se pregunta si realmente llevan los 12 artículos que cree recordar, o si son 15 porque David añadió algo a última hora. David, por su parte, intenta tachar cosas en su aplicación mientras intenta no perder el ritmo.

Esta falta de organización les genera un estrés innecesario. A veces llegan a casa y se dan cuenta de que se han dejado algo importante, y otras veces descubren que han comprado el doble de lo que necesitaban porque no se aclararon con las cantidades en el pasillo. Para intentar solucionar esto y que la compra sea algo más tranquilo, han decidido pedirte que les des un programa que les calcule cuántas cosas tenían que comprar en total y así ahorrarse más de un dolor de cabeza.

Entrada

La primera línea de entrada contendrá un entero N , el número de productos diferentes en la lista de la compra.

Las siguientes N líneas contendrán una cadena de texto (el nombre del producto, formado por un máximo de 20 letras minúsculas o mayúsculas sin espacios) y un número entero Q_i , separados por un espacio. El entero Q_i representa la cantidad que deben comprar de ese producto.

Salida

Una única línea con un número entero: el número total de artículos que comprarán.

Entrada de ejemplo

```
5
Leche 1
Huevos 1
Harina 1
Harina 1
Mantequilla 1
```

Salida de ejemplo

```
5
```

Límites

- $1 \leq N \leq 10^5$
- $0 \leq Q_i \leq 1000$ para todo $1 \leq i \leq N$

Tiempo: 0.5 segundo(s)



Hace poco se celebró en la universidad la Hack-On, un evento de ciberseguridad organizado por alumnos de los grados de Ingeniería de la Ciberseguridad de la URJC. Entre sus múltiples actividades destacaba el CTF (Capture the Flag), una competición donde los participantes deben resolver retos de diversas áreas como OSINT, Criptografía o Análisis Forense.

Los organizadores del evento son expertos en sus respectivos campos, pero parece que han dejado los algoritmos y la programación para septiembre; un curso de programación competitiva no les vendría nada mal. En uno de los retos, han configurado una máquina a la que los participantes deben acceder. El sistema está repleto de directorios y archivos, entre los cuales se esconde uno denominado “flagCTF”, que es el objetivo final.

El problema es que el creador del reto, Caasi Onazol, se entusiasmó demasiado escondiendo el archivo. Ha generado un árbol de directorios gigante y ahora teme que sea excesivamente difícil encontrar la flag entre tanta “basura” digital.

Al acceder a la máquina, los participantes solo tienen entrada a unos pocos directorios “padre” iniciales. Un detalle importante es que, debido al uso de accesos directos (enlaces simbólicos), un mismo directorio puede ser accesible desde más de un origen.

¿Podrías ayudar a Caasi calculando cuál es el número mínimo de directorios que los concursantes deben explorar para garantizar que encuentran la flag?

Entrada

La primera línea contiene dos enteros N y P . Las siguientes N líneas componen cada una la descripción de un directorio: Una palabra y un número M_i , seguido de M_i palabras. La primera palabra es el nombre del directorio descrito, mientras que las siguientes M son los ficheros o directorios que contiene. Para finalizar aparecen P líneas con el nombre de los directorios padres, aquellos a los que un concursante accede al hackear la máquina. Los nombres de los directorios se pueden componer de hasta 20 caracteres: letras mayúsculas y minúsculas de la A a la Z del alfabeto inglés y números, no aparecerán espacios ni símbolos distintos.

Salida

Por cada caso de prueba, el programa debe imprimir un único número entero: el número mínimo de pasos (o niveles de profundidad) que los concursantes deben recorrer desde cualquiera de los directorios padre para alcanzar el archivo llamado “flagCTF”.

- Si el archivo “flagCTF” se encuentra directamente dentro de un directorio padre, la distancia es 1.
- Si el archivo no es alcanzable desde ninguno de los directorios padre proporcionados, se debe imprimir “-1”.

Entrada de ejemplo

```
3 1
home 3 musica imagenes examenIP
musica 2 cancion1 cancion2
imagenes 3 perritos gatitos flagCTF
home
```

Salida de ejemplo

```
2
```

Entrada de ejemplo

```
3 2
home 3 musica examenEDA examenIP
musica 2 cancion1 cancion2
imagenes 3 perritos gatitos flagCTF
home musica
```

Salida de ejemplo

```
-1
```

Límites

- $1 \leq P < N \leq 10000$
- $1 \leq M_i \leq 100$

Tiempo: 2 segundo(s)



Enseñando a sumar

Estás enseñando a tu sobrino Bruno a sumar. Le has regalado una cajita con muchas tarjetas de números distintos, cuando sacas dos tarjetas él siempre es capaz de adivinar la suma de ambos, parece que le has enseñado bien...

A Bruno le encanta el número 10, pero se ha dado cuenta que en su cajita no hay dos tarjetas que sumen el 10 nunca, y se ha puesto un poco triste... Para la próxima vez debes asegurarte de que eso no pase.

Dada una cajita y varios números, ¿podrías decirnos si existen dos tarjetas que al sumarlas den el valor buscado?

Entrada

La primera línea contiene dos enteros N y Q . La siguiente línea tiene N números que son las tarjetas que hay en la cajita. Las siguientes Q líneas tienen un solo número, preguntando si ese número se puede conseguir con la suma de dos tarjetas distintas de la cajita.

Salida

Por cada caso de prueba número se debe imprimir *SI* si el número se puede conseguir como la suma de dos tarjetas distintas y *NO* en caso contrario.

Entrada de ejemplo

```
4 3
1 3 5 8
2
20
6
```

Salida de ejemplo

```
NO
NO
SI
```

Límites

- $1 \leq N \leq 10000$
- $1 \leq Q \leq 10000$
- La cajita no tiene números repetidos

Tiempo: 1 segundo



La ladrona de libros

El trabajo de programador es demasiado tedioso, después de acabar la carrera decidiste dejar la informática y abrir una pequeña tiendecita de libros cerca de casa. No ganas mucho dinero, pero la tranquilidad del trabajo merecía la pena.

Merecía la pena hasta que un 23 de abril tu cabeza de ingeniero empezó a hacer las cuentas de lo recaudado ese día y para tu sorpresa... ALGUIEN TE ESTABA ROBANDO!!

Manos a la obra, has instalado en tu tiendecita un sistema de cámaras, pero contratar a alguien para vigilarlas sería demasiado caro, así que has programado un sistema que te indica cuando un cliente coge un libro.

Las cestas de tu tienda son demasiado estrechas por lo que si un cliente coge varios libros deberá apilarlos de forma que solo podrá acceder al último libro que ha cogido.

El problema es que los clientes a veces se equivocan y deciden devolver el libro a la estantería, en ese caso dejarán en su sitio el último libro que hayan guardado.

Tú recibirás los libros en la caja para cobrarlos, de esa manera si no se registran en el sistema de la caja en el orden natural, es que se ha guardado uno en algún lado.

Con todos estos sucesos ¿podrás descubrir a la ladrona de libros?

Entrada

La primera línea contiene cuatro números C, K, N, L : el número de clientes que visitarán la tienda, el número K de baldas de la estantería, el número N de sucesos que ocurrirán en la tienda y L la cantidad de libros que pasarán por la caja registradora en una jornada.

La siguiente línea contendrá C palabras, cada una indicando el nombre de un cliente.

Las siguientes K líneas nos darán la descripción de la estantería. Se indicarán en el orden de baldas que libro contiene cada una, de esa manera el primer título corresponde a la primera balda, el segundo título a la segunda, etc. En tu tienda hay libros para todos, por lo que las estanterías de un mismo ejemplar no se acabarán nunca.

Las siguientes N líneas contendrán los sucesos de la tienda, existen 3 tipos de sucesos, con los siguientes formatos:

- El cliente C_i coge un libro de la balda K_i : "1 C_i K_i "
- El cliente C_i se equivoca y deja su último libro escogido: "2 C_i "
- El cliente C_i se dirige a la caja registradora y compra todos los libros de su cesta: 3 C_i

La última línea contendrá L palabras cada una indicando el nombre de un libro, ordenados de la misma manera que entran por la caja registradora.

Salida

En caso de que falte algún libro por cobrar se deberá imprimir el nombre del ladrón, si no hubo ladrones esta jornada se deberá imprimir “Hoy no ha venido la ladrona de libros”. Cuando un ladrón es identificado la tienda cerrará por lo que en caso de haber más irregularidades en la caja registradora solo se identificará al primer ladrón.

Aunque la ficción es maravillosa, tú vendes libros, no vives en ellos. Por lo tanto no aparecerán libros de la nada ni en un orden distinto al que los clientes los ponen en la caja registradora, la única irregularidad posible es que intenten robarlos.

Entrada de ejemplo

```
3 4 7 3
Cristian Sara Alicia
La_Ladrona_De_Libros
La_Guia_Del_Autoestopista_Galactico
El_Camino_De_Los_Reyes
Los_Juegos_Del_Hambre
1 Cristian 1
2 Cristian
1 Cristian 3
1 Sara 2
1 Sara 4
3 Cristian
3 Sara
El_Camino_De_Los_Reyes Los_Juegos_Del_Hambre La_Guia_Del_Autoestopista_Galactico
```

Salida de ejemplo

```
Hoy no ha venido la ladrona de libros
```

Entrada de ejemplo

```
1 4 3 1
Isaac
La_Ladrona_De_Libros
La_Guia_Del_Autoestopista_Galactico
El_Camino_De_Los_Reyes
Los_Juegos_Del_Hambre
1 Isaac 1
1 Isaac 2
3 Isaac
La_Ladrona_De_Libros
```

Salida de ejemplo

```
Isaac
```

Límites

- $1 \leq C, K, N, L \leq 20000$
- Los nombres de los libros y de los clientes están compuestos por letras mayúsculas y minúsculas del alfabeto inglés y por barras bajas (-).