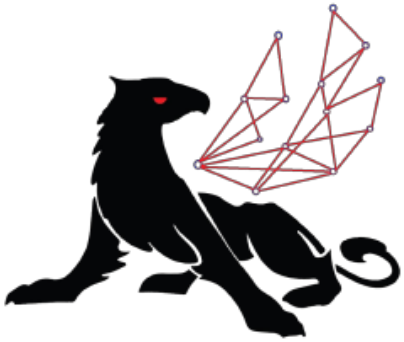

Concurso de Programación
Curso CP URJC 22/23 - FINAL

Cuadernillo de problemas

Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
28 de abril de 2023

*Stop learning useless algorithms, go and get some b*****s.*
Amir_Nuriyev

28 de abril de 2023

Índice

Autores de los problemas:

- Alicia Pina (Universidad Rey Juan Carlos)
- Cristian Pérez (Universidad Rey Juan Carlos)
- Isaac Lozano (Universidad Rey Juan Carlos)
- Raúl Fauste (Universidad Rey Juan Carlos)
- Sara García (Universidad Rey Juan Carlos)
- Sergio Salazar (Universidad Rey Juan Carlos)
- Xuqiang Liu Xu (Universidad Rey Juan Carlos)

Tiempo: 1 segundo

A
Otro aburrido problema de Balonmano

DESASTRE TOTAL!! Hace unos cuantos concursos ayudasteis a preparar un torneo de balonmano y llega la hora de entregar los premios a los ganadores. Cuando fuisteis a pedir las actas de los partidos a los árbitros, dio la casualidad de que las habían perdido.

Dentro de todo este caos Raúl os indica que él ha ido colgando la clasificación día a día en la página del club, gracias a ello podéis descargaros todas las actas parciales y observar quien fue el ganador. Dado que la pagina web se actualiza siempre al último día, se han descargado en sentido inverso al cronológico, es decir la primera clasificación corresponde al último día de competición.

Cada clasificación indica la situación del torneo al final de una jornada específica, indicando por orden de puesto de cada equipo su estado en el torneo, con las siguientes estadísticas: partidos ganados, partidos empatados, partidos perdidos, puntos actualmente, goles a favor y goles en contra.

Cada equipo jugará exactamente un partido cada jornada y el ganador será aquel que más puntos tenga al final del torneo, en caso de que varios equipos tengan el mismo número de puntos, el ganador será aquel que más goles a favor tenga, en caso de mismo número de puntos y goles a favor, el ganador será aquel con menos goles en contra.

Dada las clasificaciones por cada una de las jornadas en orden cronológico, ¿Podríais indicar quién es el ganador del torneo?

Entrada

La entrada estará compuesta de un único caso de prueba. La primera línea contiene un número N, el número de equipos. Posteriormente vendrán la descripción de la clasificación del torneo en cada una de las $N-1$ jornadas, separadas por dos lineas de 10 guiones. Cada jornada contendrá una linea indicando las estadísticas mencionadas, que siempre será : *"Equipo PG PE PP Puntos GF GC"* y posteriormente N lineas más, empezando con el nombre del equipo y seis números $PG_i, PE_i, PP_i, P_i, GF_i, GC_i$ indicando respectivamente las estadísticas mencionadas.

Salida

Por cada caso de prueba deberás imprimir el nombre del ganador del torneo, se asegura que siempre habrá un único ganador.

Entrada de ejemplo

```
4
----------
Equipo PG PE PP Puntos GF GC
Pinto 3 0 0 6 79 54
Parla 2 0 1 4 72 78
Leganes 1 0 2 2 42 66
Getasur 0 0 3 0 55 71
----------
Equipo PG PE PP Puntos GF GC
Pinto 2 0 0 4 55 31
Parla 2 0 0 4 49 33
Getasur 0 0 2 0 43 50
Leganes 0 0 2 0 21 54
----------
Equipo PG PE PP Puntos GF GC
Pinto 1 0 0 2 25 22
Parla 1 0 0 2 24 12
Getasur 0 0 1 0 22 25
Leganes 0 0 1 0 12 24
----------
```

Salida de ejemplo

```
Pinto
```

Límites

- N es par.
- $1 \leq N \leq 600$
- $0 \leq PG_i, PE_i, PP_i \leq N - 1$
- $0 \leq P_i \leq 2(N - 1)$
- $0 \leq GF_i, GC_i \leq 5000$
- Los nombres de los equipos están compuestos por letras mayúsculas y minúsculas del alfabeto inglés.

Tiempo: 1 segundo

B
¿Van tías?

Raúl se ha enterado hoy de que su grupo de amigos había quedado por la tarde.
"¿Van tías?" - ha preguntado.
Cuando sus amigos han respondido que sí, Raúl no ha tardado en prepararse para salir, queriendo llegar lo antes posible a la quedada.
Sin embargo, con las prisas, Raúl ha acabado en un laberinto. Además, como está muy nervioso, la única forma de salir que se le ha ocurrido es la regla de "la mano derecha". Esta regla consiste en situar la mano derecha en la pared que se encuentra inmediatamente a su derecha en la entrada del laberinto y seguir dicha pared, esperando así salir del laberinto.
Sabiendo que existe un camino a la salida, ¿sabrías decir si conseguirá Raúl salir del laberinto?

Entrada

La entrada comienza con una línea en la que aparece el número T de casos de prueba a procesar.
Cada caso de prueba comenzará con una línea que contendrá dos números: N y M, indicando respectivamente la altura y anchura del laberinto.
A continuación, N líneas de M caracteres describirán un mapa del laberinto. El carácter # indicará un muro, y . una celda transitable.
El laberinto tendrá siempre un muro exterior (formado por #), excepto una casilla en el muro inferior que se corresponderá con la entrada en la que se encuentra Raúl y una casilla en el muro superior, que se corresponderá con la salida del laberinto por la que debe salir.

Salida

Por cada caso de prueba, se deberá imprimir "Hoy se sale" si Raúl conseguirá salir del laberinto utilizando la regla de la "mano derecha", o "No se sale :(" en caso contrario.

Entrada de ejemplo

```
1
7 11
#########.#
#.........#
#.#########
#.##......#
#.#######.#
#.........#
#.#########
```

Salida de ejemplo

```
Hoy se sale
```

Límites

- $1 \leq \mathrm{T} \leq 10$
- $3 \leq N, M \leq 1000$

Tiempo: 1 segundo

C
Fenómenos

Hoy estabas en la cafetería de la universidad con tus amigos Marta y Sebas cuando ha aparecido vuestro profesor Luis a contaros su último gran descubrimiento: ¡El campus está lleno de túneles subterráneos!
Resulta que todas las aulas de la universidad están conectadas unas con otras a través de una serie de pasillos bajo tierra.

Aunque Luis estaba gratamente sorprendido con este hallazgo también os ha contado que estos túneles suponen un gasto para la universidad, de vez en cuando hay que pintar los túneles, cambiar las antorchas, arreglar humedades...

Vosotros tres, que sois unos fenómenos, queréis ahorrarle a la universidad unas perras y os habéis dado cuenta que muchos de esos túneles entre aulas se podrían cerrar y aún así tener siempre una manera de ir de un aula a cualquier otra (esto es importante porque claro, no queremos quitarle a Luis su nueva afición de ir de clase en clase bajo tierra).

Marta se ha dedicado a contar cuántos túneles hay en la universidad y qué dos aulas une cada uno.
Sebas los ha recorrido contando metros y antorchas para ver lo que cuesta cada uno a la universidad.
Tú eres el encargado de, gracias al trabajo de tus amigos, calcular el número máximo de euros que se puede ahorrar la universidad cerrando algunos túneles sin afectar a la conexión global de las aulas.

Entrada

La entrada consiste en varios casos de prueba.

La primera línea será un número C indicando cuantos casos de prueba vendrán a continuación.

La primera línea de cada caso contiene dos números, A y T, que indican el número total de aulas de la universidad y la cantidad de túneles, respectivamente.

Las siguientes T líneas contienen 3 números describiendo cada túnel: los 2 primeros, a_1 y a_2, indican las aulas que conecta el túnel (en cualquiera de los 2 sentidos) y el tercero e_i son los euros que cuesta mantener dicho túnel i.

Aunque no hay túneles que unan un aula consigo misma, sí que puede haber 2 aulas que estén conectadas por más de un túnel, ¡menuda gestión!

Salida

La salida deberá ser un número por caso de prueba indicando el número máximo de euros que se puede ahorrar la universidad cerrando algunos de los túneles.

Entrada de ejemplo

```
2
3 3
0 1 2
2 1 3
0 2 5
5 7
0 3 10
3 1 1
0 1 20
4 0 16
2 4 7
3 2 8
1 4 100
```

Salida de ejemplo

```
5
136
```

Límites

- $1 \leq C \leq 100$
- $2 \leq A \leq 1000$
- $0 \leq a_1 \neq a_2 \leq A - 1$
- $0 \leq e_i$
- $\sum_{i=1}^{T} e_i \leq 10^9$
- $1 \leq T \leq 100000$

Tiempo: 1 segundo

D
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 el número 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 titulo 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 mas irregularidades en la caja registradora solo se identificará al primer ladrón.

Aunque la ficción es maravillosa, tu 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 de alfabeto inglés y por barras bajas (_).

Tiempo: 0.5 segundos

E
Fuga dall'ascensore

McDonuts ha lanzado una nueva colección de vasos para el menú Donuts Meal, para obtener los vasos, además de comprar el menú, los clientes deben superar un juego llamado "Fuga dall'ascensore".
En el juego, el jugador está atrapado en un ascensor, y la única forma de salir del ascensor es conseguir un determinado número. Para conseguir ese determinado valor, cada menú ofrece una tarjeta con diferentes números. El cliente debe sumar las tarjetas con el objetivo de obtener un determinado valor en un número limitado de sumas (solo puede sumar cada tarjeta una vez y si se usan menos operaciones de las marcadas no es válido).
El juego termina si ocurre uno de los siguientes eventos: obtiene el número en el número de operaciones indicadas o no consigue el número.
Para poder ver cómo de fácil según los números es el juego, nos han pedido que implementemos un programa que dada las condiciones del juego, obtener las diferentes maneras de ganar.

Entrada

La entrada consiste en varios casos de prueba. La primera línea contiene un entero K que indica el número de casos. Cada caso tiene 2 líneas.
La primera línea contiene 3 números N, M y S, que corresponden al número de tickets, el número de sumas que podemos realizar y el número que tenemos que obtener.
La segunda línea tenemos N enteros que indican el valor de cada número.

Salida

Si por lo menos hay una forma de ganar el juego se imprime el número de formas, en el caso contrario se deberá imprimir la frase "Bloccato :(".

Entrada de ejemplo

```
2
5 2 7
1 2 4 5 2
5 2 40
1 2 4 5 2
```

Salida de ejemplo

```
2
Bloccato :(
```

Límites

- $1 \leq M \leq N \leq 40$ $1 \leq K \leq N_i \leq 60$ $1 \leq S \leq 600$

Ejemplo y Notas de Python

En el primer caso de prueba, se consigue el número 7 con las operaciones 2+5 y 5+2.
En el segundo caso de prueba lo máximo que puede conseguir es 14, inferior al número solicitado.
No se tiene en cuenta el orden en que se eligen las tarjetas.
Para recursión en Python poned recursion limit a 10^4, **sys.setrecursionlimit(10000)**.

Tiempo: 2.5 segundos

F
¡Otra vez!

A Tián se le ha vuelto a olvidar poner el problema del concurso (para variar).

Lo ha intentando hacer todo en el último momento y con los nervios y las prisas, ha desordenado todos los papeles de la asociación de programación competitiva. Esto no supondría ningún problema si no tuviésemos que entregar mañana todos los documentos "Top Secret"para que nos den "la paguita".

Queremos poder comprar sudaderas para todos, así que necesitamos que nos ayudes a programar un gestor de documentos, así, cuando nos pidan los "Top Secret", los podremos coger de un plumazo.

La información que necesitamos de los documentos es el identificador del grupo al que pertenecen y el nombre del documento. Así, cada documento viene identificado por dos cadenas de texto, ID y N.

Entrada

Cada caso de prueba T tiene dos partes; la recogida de f ficheros, y q grupos de ficheros de los que la universidad necesita nombres. La primera línea indica el número f, el número de ficheros que recogeremos. A continuación, aparecen f ficheros.

Cada fichero f_i viene determinado por dos cadenas de texto, el grupo al que pertenece (ID) y su nombre (N). Después de la recogida de ficheros, habrá un salto de línea. Por último, vienen las q peticiones de la universidad. Ya sabemos cómo se pone la burocracia a veces, así que no sabemos cuántos grupos de documentos van a necesitar. Eso sí, nos garantizan que, una vez acaben de pedirnos grupos, nos lo harán saber escribiendo "$---$". No existirá un nombre de ficheros que no exista anteriormente.

Salida

Para cada grupo de documentos que nos pidan, tenemos que imprimir los nombres de todos los documentos, en el orden en el que han sido recogidos pertenecientes a dicho grupo, separados por espacio. La información de cada grupo de ficheros está separada por un salto de línea. El nombre de cada fichero tiene que estar impreso en el mismo orden en el que han entrado.

Entrada de ejemplo

```
5
TopSecret trapicheos
Public siguienteTweet
TopSecret identidadEnigma
Buro firmas
Public linkTelegram

TopSecret
Public
---
```

Salida de ejemplo

```
trapicheos identidadEnigma
siguienteTweet linkTelegram
```

Límites

- $1 \leq f \leq 1000000$ $0 \leq q \leq 1000$ $1 \leq T \leq 1000$
- ID y N están compuestos por letras mayúsculas y minúsculas del alfabeto inglés (de tamaño < 20).

Tiempo: 1 segundo

G
URJC Dates

Estamos ya en el último concurso del curso de programación competitiva por lo que va a ser un día triste para los organizadores... Sin embargo, a Raúl se le ha ocurrido una idea.
Durante el curso Raúl ha percibido que las feromonas (hormonas del amor) estaban a la orden del día en las clases del curso. Por lo que para ayudar a todos los implicados en conquistar a otro asistente/a del curso Raúl quiere diseñar una aplicación.
Pese a ser uno de los profesores del curso, a Raúl no se le da muy bien programar, por lo que va a necesitar tu ayuda para realizar dicha aplicación. La aplicación va a consistir en lo siguiente: Dos grupos de personas: el grupo de conquistadores y el grupo de conquistados. Cada persona irá puntuada con un número que denota su belleza. La finalidad de la aplicación es emparejar a las personas del grupo de conquistadores con el grupo de conquistados de forma que la penalización sea la menor posible.
¿Qué es la penalización? La penalización se refiere a la suma de las diferencias (en valor absoluto) entre las parejas formadas. Por lo que queremos minimizar esa suma. Veámoslo con un ejemplo: Si tenemos dos grupos $G_1 = \{(Ana, 9), (Pedro, 4)\}$ y $G_2 = \{(Juan, 2), (Sergio, 8)\}$. La mejor solución sería Ana-Sergio y Pedro-Juan con una penalización de 3 ya que la otra solución Ana-Juan y Pedro-Sergio tiene una penalización de 11.
Como se puede observar en el ejemplo, no hay que formar parejas Hombre-Mujer, cualquier tipo de pareja es válida. Lo importante es minimizar la penalización para que Raúl este contento. ¿Puedes ayudarle?

Entrada

La primera línea contiene un número N, el número de personas por cada grupo.
La segunda línea contiene N pares de NOMBRE P, que representa que la persona con ese nombre tiene esa puntuación P (siendo P número entero). Estas personas pertenecen al grupo de conquistadores.
La tercera línea contiene N pares de NOMBRE P, que representa que la persona con ese nombre tiene esa puntuación P (siendo P número entero). Estas personas pertenecen al grupo de conquistados.
Además, los nombres estarán formados al menos una letra mayúsculas y minúsculas del alfabeto inglés y tendrán una longitud menor a 10 letras.

Salida

Por cada caso de prueba deberás imprimir la penalización obtenida al emparejar de forma óptima.

Entrada de ejemplo

```
4
Raul 10 Alicia 4 Sergio 9 Sara 7
Isaac 9 Paco 6 Cristina 3 Maria 8
```

Salida de ejemplo

```
4
```

Límites

- $1 \leq N \leq 10^5$
- $0 \leq P \leq 10$