Concurso de Programación

Clasificatorio URJC - AdaByron 2026

http://www.ada-byron.es

Cuadernillo de problemas

URJC Escuela Técnica Superior de Ingeniería Informática

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

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

20 de marzo de 2026

Índice

A URJC Dates **3**

B Asignación de salas en congresos **5**

C ¿Cuántos números? **7**

D Raíces felices **9**

E Mesita de noche **11**

Autores de los problemas:

- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Sergio Salazar Cárdenas (Universidad Rey Juan Carlos)
- Raúl Fauste Polo (Universidad Rey Juan Carlos)
- Sara García Rodríguez (Universidad Rey Juan Carlos)

Probadores de los problemas:

- Iván Penedo Ventosa (Habla Computing)
- Lucas Martín García (Universidad Rey Juan Carlos)

Tiempo: 1 segundo

A
URJC Dates

Durante el curso Raúl ha percibido que las feromonas (hormonas del amor) estaban a la orden del día en las clases. 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

$$G_1 = \{(Ana, 9), (Pedro, 4)\} \quad \text{y} \quad G_2 = \{(Juan, 2), (Sergio, 8)\}$$

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.
Se asegura que ningún nombre será repetido. Es decir, no habrá dos personas con el mismo nombre. Además, los nombres estarán formados por letras 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

- $1 \leq N \leq 100$
- $0 \leq P \leq 10$

Tiempo: 1 segundo

B
Asignación de salas en congresos

Una conferencia tiene N presentaciones que deben planificarse en salas disponibles. Cada sala puede utilizarse durante un máximo tiempo. Además, cada presentación tiene una duración específica medida en unidades de tiempo. Una presentación debe asignarse completamente a una única sala. Tu tarea es determinar si es posible asignar todas las presentaciones a las salas disponibles.

Entrada

La primera línea contiene tres enteros N, M y K, que representan el número de presentaciones, el número de salas disponibles y el número de unidades de tiempo por sala, respectivamente.

La segunda línea contiene N enteros separados por espacios, donde cada valor representa la duración en franjas de cada presentación.

Salida

Si es posible asignar todas las presentaciones, imprime Posible en otro caso, devuelve No es posible.

Entrada de ejemplo

```
4 2 10
6 5 3 6
```

Salida de ejemplo

```
No es posible
```

Entrada de ejemplo

```
5 2 10
7 4 3 3 3
```

Salida de ejemplo

```
Posible
```

Límites

- $1 \leq N \leq 18$
- $1 \leq M \leq 10$
- $1 \leq K \leq 100$

Tiempo: 1 segundo

C
¿Cuántos números?

Tu amigo Marcos siempre te vacila por estudiar matemáticas. "¿Para qué sirve saber tantas fórmulas si luego no puedes resolver nada práctico?", te dice entre risas. Un día, harto de presumir, te lanza un reto delante de todos: "A ver, señor matemático... ¿cuántos dígitos tiene el número 2^{1000}? ¡Y rápido!" Tú sonríes. Lo resuelves en segundos. Marcos, incrédulo, decide que si no puede ganarte con uno, te abrumará con miles de preguntas del mismo estilo. Ahora te manda una lista enorme de exponentes y quiere que le respondas a todos. ¿Seguirás siendo tan listo?

Entrada

La primera línea contiene un entero T, el número de casos de prueba. Cada una de las siguientes T líneas contiene un entero N, representando el exponente de la potencia 2^N.

Salida

Para cada caso de prueba, imprime en una línea el número de dígitos que tiene 2^N.

Entrada de ejemplo

```
3
1
10
1000
```

Salida de ejemplo

```
1
4
302
```

Límites

- $1 \leq T \leq 10000$
- $1 \leq N \leq 10^8$

Tiempo: 0.5 segundo

D
Raíces felices

Un número x se dice que es una raíz feliz si existe otro número y de forma que $\sqrt{y} = x$. En este problema te pido que, dado un número n, me devuelvas las n primeras raíces felices, ordenadas de menor a mayor.

Entrada

La entrada comienza con un número$c$, el número de casos de prueba. A continuación aparecen c líneas, cada una con un número n mencionando cuántas raíces felices deben aparecer.

Salida

Por cada caso de prueba imprime las n primeras raíces felices, separadas por un espacio, ordenadas de menor a mayor.

Entrada de ejemplo

```
2
1
3
```

Salida de ejemplo

```
1
1 2 3
```

Límites

- $1 \leq c, n \leq 10^5$

Tiempo: 0.5 segundos

E
Mesita de noche

Este año me he propuesto leer más. Cuando veo un libro que me interesa me lo compro y comienzo a leerlo. Suelo ponerme a leer antes de dormir, así que voy dejando los libros que me estoy leyendo en la mesita de noche. Cuando he terminado un libro, lo quito de la mesita. El problema es que a veces no me he acabado un libro y empiezo a leerme otro. La montaña de libros que se acaba formando en mi mesita es bastante alta.

Cada noche pueden pasar 2 cosas:

1. Llego con un libro nuevo y este pasa a ser el que me estoy leyendo, el más alto en mi montaña de libros.

2. Me entra la vena lectora y me acabo uno, o incluso varios, de los libros que me estoy leyendo, quitándolos de la mesita. El orden de lectura que sigo es siempre del más nuevo en mi mesita al más viejo.

Como el techo de mi casa no es infinito, necesito saber cuántos libros quedan en mi mesita de noche y cuáles son.

Entrada

La entrada corresponde a un único caso de prueba. La primera línea es un entero N, que indica el número de noches. Le siguen N líneas con un número entero e_i, que puede ser 1 o 2. Cuando e_i es 1, significa que he comprado un libro nuevo. El título del nuevo libro, t_i, aparece a continuación tras un espacio y consiste en un único string. Cuando e_i es 2, significa que me he leído algunos libros. El número de libros que me leo y quito de la mesita, l_i, aparece a continuación tras un espacio.

Salida

La salida consistirá en un entero M indicando cuántos libros quedan al final en la mesita de noche. A continuación deben aparecer M líneas, cada una de ellas indicando el título de un libro que queda en la mesita. Los libros deben aparecer en el orden que están en mi montaña, de más nuevo a más viejo.

Entrada de ejemplo

```
5
1 LaLadronaDeLibros
1 NoTodoElMundo
1 LosNombresPropios
2 2
1 Oxigeno
```

Salida de ejemplo

```
2
Oxigeno
LaLadronaDeLibros
```

Límites

- $1 \leq N \leq 10^6$
- $n_i \in \{1, 2\}$ con $i = 1, ..., N$
- t_i es una única cadena de caracteres del alfabeto inglés, sin espacios, con $i = 1, ..., N$.
- $1 \leq l_i \leq$ nº libro en la mesita con $i = 1, ..., N$