



# Concurso de Programación Concurso Final 2025

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

## Cuadernillo de problemas



Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**  
11 de abril de 2025

# Índice

<b>A Microondas</b>	<b>3</b>
<b>B Bonsai</b>	<b>4</b>
<b>C Anchas Escaleras</b>	<b>5</b>
<b>D Envíos Prioritarios</b>	<b>6</b>
<b>E Una ronda más</b>	<b>7</b>
<b>F ¡Saca tú!</b>	<b>8</b>
<b>G A por bollos</b>	<b>9</b>

Autores de los problemas:

- Raúl Fauste Jimenez (Universidad Rey Juan Carlos)
- Sara García Rodríguez (Universidad Rey Juan Carlos)
- Lucas Martín García (Universidad Rey Juan Carlos)
- Alicia Pina Zapata (Universidad Rey Juan Carlos)

*Testers* no autores de los problemas:

- Adaya Ruiz Mayoral (GMV, Universidad Rey Juan Carlos)
- Alejandro Mayoral Gómez (Universidad Rey Juan Carlos)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Sergio Salazar Cardenas (Universidad Rey Juan Carlos)
- David Enrique Orna Alcobendas (Universidad Rey Juan Carlos)
- Eva Gómez Fernández (Universidad Rey Juan Carlos)
- Iván Penedo Ventosa (Universidad Rey Juan Carlos)

Tiempo: 0.5 segundos

# A

## Microondas

En la cafetería de mi universidad hay un enorme problema, casi todos los microondas están sucios. De hecho he llegado hoy a la hora de comer a calentar todos los tupperes que traía de casa y resulta que solo hay un microondas que esté limpio. Voy a usar solo ese, prefiero no comer a arriesgarme a una intoxicación.

Además, tengo otro problema, no sé si voy a poder calentarlos todos. Cada tupper requiere un tiempo para calentarse y yo solo estoy dispuesto a tirarme calentando un cierto tiempo, todo el mundo quiere usar ese microondas y yo no quiero que me mire mal toda la universidad.

Por otro lado, no todos los tupperes que me he traído me gustan por igual. Puedo ponerles a cada uno una nota. Sabiendo que tengo un tiempo límite de uso del microondas y que cada tupper necesita un tiempo para calentarse y yo lo he valorado con cierta puntuación, ¿podrías decirme cuál es la suma de puntuaciones máxima que puedo obtener?

Nótese que como solo hay un microondas, solo puedo calentar un tupper a la vez, pero no hay que esperar ningún tiempo entre tupperes.

### Entrada

La entrada consistirá en un único caso. Comenzará con una línea con dos números separados por un espacio,  $T$  y  $N$ .  $T$  es el tiempo máximo que quiero usar el microondas y  $N$  es el número de tupperes que traigo. A continuación, aparecerán  $N$  líneas con dos números, el primero  $t_i$  indicando el tiempo que tarda ese tupper en calentarse, y el segundo  $p_i$  indicando la puntuación que le doy.

### Salida

La salida será una única línea, un número indicando la suma máxima de puntuaciones que puedo obtener.

### Entrada de ejemplo

```
60 4
50 7
20 5
60 10
40 6
```

### Salida de ejemplo

```
11
```

### Límites

- $1 \leq N \leq 100$
- $1 \leq T \leq 5000$
- $1 \leq t_i \leq 100$
- $1 \leq p_i \leq 10000$

Tiempo: 0.4 segundos

## ● B Bonsai

Mi amigo Sergio se acaba de unir a un club de jardinería. El otro día, mientras esperábamos a que acabase el concurso de Ada Byron, me estuvo contando algunas curiosidades que había aprendido. Me dijo que las raíces que una planta es capaz de producir dependen directamente de la cantidad de agua que ha recibido a lo largo de su vida.

Concretamente, si una planta ha sido regada con una cantidad  $e$  de litros de agua a lo largo de su vida, el número de raíces que generará viene determinado por  $R = e^2 - e$ .

A mí, me gusta considerarme una persona observadora, cada vez que salgo a dar un paseo y veo una secuencia de plantas en el balcón de un piso, me empiezo a hacer preguntas: ¿Cuántas raíces habrá entre la segunda y la quinta planta? ¿Cuántas raíces habrá en total?

### Entrada

La entrada se compone de un único caso de prueba. La primera línea contiene dos números enteros,  $N$ , el número de plantas en el balcón y  $M$ , el número de consultas. La siguiente línea se compone de  $N$  números  $e_i$  representando los litros de agua utilizados para regar la planta en la posición  $i$ . Las siguientes  $M$  líneas son consultas de dos tipos:

- $1 \ i \ v$ : Donde se actualiza el número de litros de agua utilizado para regar la planta en la posición  $i$  al valor  $v$ .
- $2 \ l \ r$ : Donde se pregunta por el total de raíces generadas por las plantas contenidas en el intervalo que comienza por la planta en la posición  $l$  y termina con la planta en la posición  $r - 1$ .

### Salida

Para cada consulta del segundo tipo se debe imprimir el total de raíces generadas en cada intervalo.

### Entrada de ejemplo

```
5 5
5 4 2 3 5
2 0 3
1 1 1
2 0 3
1 3 1
2 0 5
```

### Salida de ejemplo

```
34
22
42
```

### Límites

- $1 \leq N \leq 10^5$        $1 \leq M \leq 2 \cdot 10^4$        $1 \leq e_i \leq 10^4$
- $0 \leq i < N$        $1 \leq v \leq 10^4$        $0 \leq l < r \leq N$

Tiempo: 0.5 segundo



## Anchas Escaleras

No sé si te has fijado, pero hay escaleras que son más anchas de lo normal. Por ejemplo, las que van de los aularios a la cafetería de este campus.

Como son un poco incómodas, uno nunca sabe la mejor manera de bajarlas: de 1 en 1, de 2 en 2, alternando 1 y 2... Hay muchas maneras de bajarlas y me ha entrado la curiosidad de saber exactamente cuántas.

Te voy a decir el número total de escalones que hay en ciertas escaleras y cuántos escalones sé bajar de golpe. Teniendo en cuenta que no te puedes pasar del último escalón, ¿cuántas maneras hay de bajar?

### Entrada

La entrada consiste en un único caso de prueba. En la primera línea hay un entero  $E$ , indicando el número total de escalones, y otro entero  $G$ , con el número de diferentes escalones que sé subir de golpe. En la siguiente línea aparecen los  $G$  números  $g_i$ , todos distintos entre sí, que indican cuantos escalones sé bajar de golpe.

### Salida

Se debe imprimir un único entero, las maneras de bajar las escaleras. Como este número puede ser muy grande, da la solución módulo **1000000007**.

### Entrada de ejemplo

```
3 2
1 4
```

### Salida de ejemplo

```
1
```

### Entrada de ejemplo

```
5 2
3 1
```

### Salida de ejemplo

```
4
```

### Entrada de ejemplo

```
5 3
4 2 6
```

### Salida de ejemplo

```
0
```

### Límites

- $1 \leq E \leq 100$
- $1 \leq G \leq 10$
- $1 \leq g_i \leq 100$

Tiempo: 1 segundo

# ● D

## Envíos Prioritarios

Amazon está pensando añadir un nuevo tipo de suscripción a su servicio de compras online. El nuevo servicio lleva las siglas de “P2W”, nadie está seguro del significado que llevan detrás. Los usuarios que opten por pagar los 100 euros que cuesta podrán hacer que sus envíos se salten posiciones en la cola de salida de los centros de reparto; en concreto, saldrán antes que cualquier producto que no se haya pedido con este servicio.

### Entrada

La entrada contendrá un único caso de prueba. El caso empezará con un número,  $N$ , que indicará el número de eventos que sucederán a lo largo de una semana en el centro de reparto. Cada evento ocupará una línea y puede tener uno de los siguientes formatos:

- “P2W” o “NOR” - indicando la llegada de un paquete “P2W” o normal a la cola de reparto.
- “R C” - indicando que va a salir un camión con capacidad para  $C$  paquetes del centro de reparto.

Tanto en la cola de reparto como en el camión los paquetes se ordenan por ser “P2W” o no y luego por su orden de aparición.

### Salida

Por cada camión que sale tienes que imprimir los identificadores de los paquetes que se va a llevar el camión (sin rebasar su capacidad), separados por un espacio. El camión puede salir sin llenarse e incluso vacío si no hay paquetes suficientes esperando en la cola. Los paquetes se identifican por su orden de aparición en la entrada, empezando en el 0.

### Entrada de ejemplo

```
8
NOR
NOR
P2W
NOR
R 2
P2W
R 5
NOR
```

### Salida de ejemplo

```
2 0
4 1 3
```

### Límites

- $1 \leq N \leq 200000$
- $1 \leq C \leq 100$

Tiempo: 1 segundo



## Una ronda más

En el curso de programación competitiva, los profesores tienen una curiosa tradición: después de cada clase, se dirigen a la refresquería local para disfrutar de unos refrigerios y debatir sobre estructuras de datos y complejidades algorítmicas. Cada uno de ellos pide siempre un tipo distinto de refresco, y consume una cantidad variable. Eso sí, son tan meticulosos que siempre apuntan dos cosas: la cantidad de refrescos que han tomado y el precio de su refresco. Bueno... casi todos.

Por algún extraño fenómeno, siempre hay exactamente un profesor que se olvida del precio de su refresco. Al terminar la jornada de debate, el refresquero siempre les da la cuenta con el precio total pero sin los precios individuales.

Tu misión es ayudarles a calcular cuánto debe pagar el profesor olvidadizo, usando la información de los demás. En cada ocasión acude un número variable de profesores, pero siempre exactamente uno de ellos ha olvidado el precio unitario de su bebida.

### Entrada

La entrada consistirá en distintos casos de prueba. Cada caso de prueba comienza con un número  $N$  que indica el número de profesores que han acudido a la refresquería. En la siguiente línea aparecen  $N$  líneas, una por cada profesor, que contienen dos números enteros positivos  $C_i$  y  $P_i$ , indicando correspondientemente la cantidad de refrescos que ha tomado el profesor  $i$  y el precio del refresco. En el caso del profesor olvidadizo, en lugar del precio  $P_i$ , aparecerá el valor  $-1$  indicando que ese es el precio que se desconoce. Por último, en la última línea aparecerá el precio total de la cuenta proporcionado por el refresquero. La entrada terminará con un 0, que marcará el final de la entrada y no generará salida.

### Salida

Por cada caso se debe imprimir el precio del refresco del profesor olvidadizo.

### Entrada de ejemplo

```
3
2 5
3 -1
1 10
35
2
4 -1
3 6
26
0
```

### Salida de ejemplo

```
5
2
```

### Límites

- $1 \leq N \leq 10^3$ ,  $1 \leq C_i \leq 10^3$ ,  $1 \leq P_i \leq 10^3$

Tiempo: 1 segundo

# F ¡Saca tú!

En una variante del tenis de mesa (ping-pong), el saque debe realizarse alternando el lado de la mesa en cada punto, sin importar qué jugador saca. Nada más empezar, cuando el marcador está a 0 para ambos jugadores, se debe sacar desde el lado derecho, el siguiente punto, el segundo, desde el izquierdo, el tercero, desde el derecho nuevamente, y así sucesivamente.

Dado el marcador actual del juego, tu misión es determinar desde qué lado debe realizarse el siguiente saque.

## Entrada

La entrada comienza por el número  $N$  de casos de prueba que vendrán a continuación. Cada caso consiste en una única línea conteniendo dos enteros  $A$  y  $B$  separados por un espacio, que representan la cantidad de puntos ganados por el jugador 1 y el jugador 2, respectivamente.

## Salida

Para cada caso de prueba se debe imprimir una única línea con la palabra DERECHA o IZQUIERDA, indicando desde qué lado debe realizarse el siguiente saque.

## Entrada de ejemplo

```
5
0 0
1 0
14 8
35 3
2 5
```

## Salida de ejemplo

```
DERECHA
IZQUIERDA
DERECHA
DERECHA
IZQUIERDA
```

## Límites

- $1 \leq N \leq 10^6$
- $0 \leq A, B \leq 10^6$

Tiempo: 0.25 segundo

# G

## A por bollos

Ya van 3 horas de AdaByron y sinceramente a estas alturas de concurso solo puedes pensar en qué pondrán para comer. Este año han dicho que no van a poner bollos en el pasillo, pero, como no tienes nada mejor que hacer, has decidido salir a ver si cae algo.

Al salir al pasillo te has encontrado con una sala bastante laberíntica y estás convencido de que al otro lado están los coches, zampándose los bollos que tanto deseas. Aunque de primeras no ves la salida, estás convencido de que existe y, como te sobra mucho tiempo, decides tomar la estrategia de seguir la pared de la izquierda, esperando llegar a la salida en algún momento. ¿Conseguirás llegar a la salida?

### Entrada

La entrada comienza con 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 de la sala. A continuación,  $N$  líneas de  $M$  caracteres describirán un mapa de la misma. El carácter `#` indicará una pared, y `.` una celda transitable. La sala contará siempre con una pared exterior (formado por `#`), excepto por una casilla en la pared inferior, que se corresponderá con la entrada en la que te encuentras, y una casilla en la pared superior, que se corresponderá con la salida de la sala.

### Salida

Por cada caso de prueba, se deberá imprimir 'Hay bollos' si conseguirás salir de la sala, siguiendo la estrategia descrita, o 'No hay bollos :(' si nunca saldrás de ella.

### Entrada de ejemplo

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

### Salida de ejemplo

```
Hay bollos
```

### Límites

- $1 \leq T \leq 10$
- $3 \leq N, M \leq 10000$