



## Olimpiada Chilena de Informática 2023

*11 de Enero, 2023*

*Las siguientes personas participaron en la elaboración de este conjunto de problemas:*



## Información General

Esta página muestra información general que se aplica a todos los problemas.

## Envío de una solución

1. Los participantes deben enviar **un solo archivo** con el código fuente de su solución.
2. El nombre del archivo debe tener la extensión `.cpp` o `.java` dependiendo de si la solución está escrita en **C++** o **Java** respectivamente. Para enviar una solución en Java hay que seguir algunos pasos adicionales. Ver detalles más abajo.

## Casos de prueba, subtareas y puntaje

1. La solución enviada por los participantes será ejecutada varias veces con distintos casos de prueba.
2. A menos que se indique lo contrario, cada problema define diferentes subtareas que lo restringen. Se asignará puntaje de acuerdo a la cantidad de subtareas que se logre solucionar de manera correcta.
3. A menos que se indique lo contrario, para obtener el puntaje en una subtarea se debe tener correctos todos los casos de prueba incluidos en ella.
4. Una solución puede resolver al mismo tiempo más de una subtarea.
5. La solución es ejecutada con cada caso de prueba de manera independiente y por tanto puede fallar en algunas subtareas sin influir en la ejecución de otras.

## Entrada

1. Toda lectura debe ser hecha desde la **entrada estándar** usando, por ejemplo, las funciones `scanf` o `std::cin` en C++ o la clase `BufferedReader` en Java.
2. La entrada corresponde a un solo caso de prueba, el cual está descrito en varias líneas dependiendo del problema.
3. **Se garantiza que la entrada sigue el formato descrito** en el enunciado de cada problema.

## Salida

1. Toda escritura debe ser hecha hacia la **salida estándar** usando, por ejemplo, las funciones `printf`, `std::cout` en C++ o `System.out.println` en Java.
2. El formato de salida es explicado en el enunciado de cada problema.
3. **La salida del programa debe cumplir estrictamente con el formato indicado**, considerando los espacios, las mayúsculas y minúsculas.
4. Toda línea, incluyendo la última, debe terminar con un salto de línea.

## Envío de una solución en Java

1. Cada problema tiene un *nombre clave* que será especificado en el enunciado. Este nombre clave será también utilizado en el sistema de evaluación para identificar al problema.
2. Para enviar correctamente una solución en Java, el archivo debe contener una clase llamada igual que el nombre clave del problema. Esta clase debe contener también el método `main`. Por ejemplo, si el nombre clave es `marraqueta`, el archivo con la solución debe llamarse `marraqueta.java` y tener la siguiente estructura:

```
public class marraqueta {  
    public static void main (String[] args) {  
        // tu solución va aquí  
    }  
}
```

3. Si el archivo no contiene la clase con el nombre correcto, el sistema de evaluación reportará un error de compilación.
4. La clase no debe estar contenida dentro de un *package*. Hay que tener cuidado pues algunos entornos de desarrollo como Eclipse incluyen las clases en un *package* por defecto.
5. Si la clase está contenida dentro de un *package*, el sistema reportará un error de compilación.

## Problema A

### Suma de ejemplo

*nombre clave:* balancin

Este es un problema de ejemplo para mostrar cómo usar ocimatic. El problema es muy simple. Te dan dos enteros y tienes que imprimir su suma. Pero ten cuidado, ¡la suma podría no caber en un entero de 32 bits con signo!

#### Entrada

La entrada consiste en una sola línea con dos enteros  $a$  y  $b$  ( $-2 \cdot 10^9 \leq a, b \leq 2 \cdot 10^9$ ).

#### Salida

La salida debe contener un único entero correspondiente a la suma de  $a$  y  $b$ .

#### Subtareas y puntaje

##### Subtarea 1 (50 puntos)

Se probarán varios casos de prueba donde  $-10^9 \leq a, b \leq 10^9$ .

##### Subtarea 2 (50 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

#### Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
10 20	30

Entrada de ejemplo	Salida de ejemplo
-1 3	2



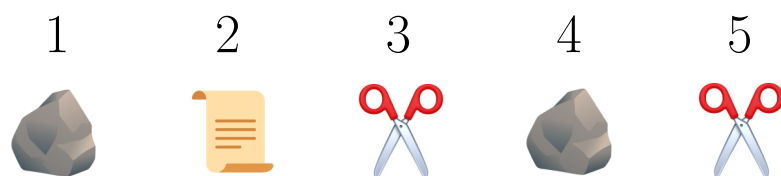
## Problema B

### Suma de ejemplo

*nombre clave:* cachipun

Finalmente ha llegado el día que todos estaban esperando: la fase final de la Olimpiada de Cachipun Internacional (OCI). En esta competencia,  $n$  oponentes se enfrentan en  $k$  rondas de combate todos-contrataodos, donde en cada ronda, cada participante elige una opción entre piedra, papel y tijera. Cada jugador gana 1 punto por cada contrincante que venció, pierde 1 por cada contrincante que lo venció, y los empates no afectan su puntuación.

Por ejemplo, en una ronda con 5 jugadores, donde tú eres el jugador 1 y los jugadores juegan de la siguiente forma:



Tú le ganarías a los jugadores 3 y 5, obteniendo 2 puntos, pierdes contra el jugador 2 que resta 1 punto, y empatas con el jugador 4, para un resultado total de 1 punto a tu favor.

Normalmente, el registro de los puntos se maneja utilizando el Cachipun Management System (CMS), sin embargo, Diego encontró una vulnerabilidad en el software, y como no hay tiempo suficiente para arreglarlo, la administración decidió pedirte a ti que crees un programa que lo sustituya.

Quedan menos de 4 horas para que comience la competencia, ¿podrás salvar la OCI?

### Entrada

Te dan  $n$  ( $2 \leq n \leq 1000$ ),  $k$  ( $1 \leq k \leq 1000$ ) y la matriz de  $k \times n$ , donde la  $i$ -ésima fila representa la  $i$ -ésima ronda, y la  $j$ -ésima columna representa la jugada del  $j$ -ésimo jugador. El carácter R representa roca, P representa papel y T tijeras.

### Salida

Debes imprimir  $n$  números, donde el  $i$ -ésimo número corresponde a la puntuación final del jugador  $i$ .

### Subtareas y puntaje

#### Subtarea 1 (20 puntos)

Se probarán varios casos de prueba donde  $n = 2$ , es decir, hay exactamente 2 jugadores.

#### Subtarea 2 (40 puntos)

Se probarán varios casos de prueba donde  $n \leq 100$ .

### Subtarea 3 (40 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

#### Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
5 1 R P T R T	1 0 -1 1 -1

Entrada de ejemplo	Salida de ejemplo
3 3 P T T R P T P P R	-1 2 -1



## Problema C

### Suma de ejemplo

*nombre clave:* dado

Este es un problema de ejemplo para mostrar cómo usar ocimatic. El problema es muy simple. Te dan dos enteros y tienes que imprimir su suma. Pero ten cuidado, ¡la suma podría no caber en un entero de 32 bits con signo!

#### Entrada

La entrada consiste en una sola línea con dos enteros  $a$  y  $b$  ( $-2 \cdot 10^9 \leq a, b \leq 2 \cdot 10^9$ ).

#### Salida

La salida debe contener un único entero correspondiente a la suma de  $a$  y  $b$ .

#### Subtareas y puntaje

##### Subtarea 1 (50 puntos)

Se probarán varios casos de prueba donde  $-10^9 \leq a, b \leq 10^9$ .

##### Subtarea 2 (50 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

#### Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
10 20	30

Entrada de ejemplo	Salida de ejemplo
-1 3	2



## Problema D

### Suma de ejemplo

*nombre clave:* **figuras**

Este es un problema de ejemplo para mostrar cómo usar ocimatic. El problema es muy simple. Te dan dos enteros y tienes que imprimir su suma. Pero ten cuidado, ¡la suma podría no caber en un entero de 32 bits con signo!

#### Entrada

La entrada consiste en una sola línea con dos enteros  $a$  y  $b$  ( $-2 \cdot 10^9 \leq a, b \leq 2 \cdot 10^9$ ).

#### Salida

La salida debe contener un único entero correspondiente a la suma de  $a$  y  $b$ .

#### Subtareas y puntaje

##### Subtarea 1 (50 puntos)

Se probarán varios casos de prueba donde  $-10^9 \leq a, b \leq 10^9$ .

##### Subtarea 2 (50 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

#### Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
10 20	30

Entrada de ejemplo	Salida de ejemplo
-1 3	2

