Olimpiada Chilena de Informática 2024

11 de Enero, 2024

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 incluídos 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

Apuesta

nombre clave: `apuesta`

Fernanda y tú decidieron hacer una apuesta de alto riesgo: Fernanda lanza una moneda. Si cae sello, pasarán a la Industria Chilena de Papas Caseras (ICPC) a comerse una porción de papas, de lo contario, se quedarán de brazos cruzados todo el día. Lamentablemente, la moneda cayó cara, por lo que el destino decidió que se perderían de esta increíble oportunidad. Pero Fernanda, no contenta con el resultado, decidió hacer otra apuesta.

Fernanda lanza n monedas en una fila. Si en la mayoría estricta de segmentos[1] de la fila, la mayoría de las monedas caen sello, irán a comer papas de todas formas.

Una vez que lanzó las monedas, Fernanda se puso a contar la cantidad de segmentos exitosos y totales utilizando sus conocimentos avanzados de combinatoria.

Sin embargo, le está tomando demasiado tiempo y la ICPC cierra en menos de 4 horas, por lo que decides sacas tu computador y te pones a programar una solución.

Entrada

La primera linea contiene un único entero n ($1 \leq n \leq 10^6$), la cantidad de monedas.

La segunda línea contiene n enteros m_i, el resultado de la i-ésima moneda de la fila. m_i es -1 si la moneda cayó sello, y 1 en caso de haber caido cara.

Salida

La salida debe contener dos enteros a, b, que corresponden respectivamente a la cantidad de segmentos donde la mayoría estricta de monedas es sello, y la cantidad total de segmentos.

Subtareas y puntaje

Subtarea 1 (10 puntos)
Se probarán varios casos de prueba donde $n \leq 4$

Subtarea 2 (20 puntos)
Se probarán varios casos de prueba donde $n \leq 100$ (valor referencial, $O(n^3)$ debería funcionar)

Subtarea 3 (30 puntos)
Se probarán varios casos de prueba donde $n \leq 10^4$ (valor referencial, $O(n^2)$ debería funcionar)

[1] Un segmento es un arreglo contiguo de monedas. Por ejemplo, todas las monedas desde la tercera hasta la quinta.

Subtarea 4 (40 puntos)

Se probarán varios casos de prueba sin restricciones adicionales. (10^6 como valor referencial, $O(nlogn)$ debería funcionar)

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3 1 -1 1	3 6

Entrada de ejemplo	Salida de ejemplo
5 1 1 1 1 1	10 10

Problema B

Suma de ejemplo

nombre clave: `baldosas`

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 C
Suma de ejemplo

nombre clave: `noctulo`

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: `tranvia`

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`