



## Aviso

Si usted no puede enviar una solución a través del sistema, por favor haga lo siguiente:

- Coloque los archivos que quiere enviar en una carpeta llamada `submit_<nombre de la tarea>` en el escritorio (el directorio ya debería existir) **antes que la competencia finalice**.
  - Para problemas de solo-salida, sus archivos deben llamarse `01.out`, `02.out`, ..., `10.out`.
  - Para otros problemas, debe haber **exactamente un** archivo en esa carpeta, conteniendo su código fuente.
- Pídale al líder de su equipo que presente un reclamo.

Si usted piensa que debería tener tiempo extra, por favor haga lo siguiente:

- Envíe un pedido de aclaración (en papel o usando el sistema de la competencia) tan pronto como sea posible.
- No deje su escritorio ni hable con otros participantes después que termine la competencia.

Cada problema tiene un paquete adjunto que usted puede descargar del sistema de la competencia.

Hay algunos problemas de "solo-salida", para los cuales:

- El paquete adjunto contiene los casos de entradas, tanto los del problema como los de ejemplo. Cada caso de prueba es una subtarea diferente.
- Usted puede enviar múltiples archivos como una carpeta comprimida (zip file). Para este propósito, sus archivos de salida deben ser llamados `?? .out`, donde `??` es el número de caso de prueba (por ejemplo, `03.out`). Usted puede comprimir múltiples archivos utilizando el siguiente comando: `zip output.zip *.out`
- Usted puede hacer hasta 100 envíos para los problemas de solo-salida. En cada envío, usted puede enviar los archivos de salida para cualquier subconjunto de casos de prueba.

Para los otros problemas:

- El paquete adjunto contiene calificadores de ejemplo, implementaciones de ejemplo, casos de prueba de ejemplo, y comandos de compilación.
- Usted puede hacer hasta 50 envíos para cada problema, y tiene que enviar exactamente un archivo en cada envío.
- El nombre del archivo que usted debe enviar está dado en el encabezado del

enunciado del problema. El archivo debe implementar las funciones descritas en el enunciado del problema utilizando los tipos de dato que se encuentran en las implementaciones de ejemplo.

- Usted es libre de implementar otras funciones.
- Sus envíos no deben leer de entrada estándar, escribir en salida estándar, o interactuar con cualquier otro archivo. Sin embargo, se puede escribir en la salida estándar de error.
- Sus envíos **no deben llamar a `exit()` o a `System.exit()`**
- Probando sus programas con el calificador de prueba, su entrada debe coincidir con el formato y las restricciones del enunciado del problema, de otra manera, comportamientos no especificados podrían ocurrir.
- En las entradas de ejemplo, cada dos tokens consecutivos en una línea son separados por un solo espacio, a menos que se especifique explícitamente otro formato.
- Cuando pruebe su código en su computadora local, le recomendamos utilizar los comandos de compilación que se encuentran en el paquete adjunto. De otra manera, especialmente en C++, asegúrese de añadir la opción `-std=gnu++14` para compilar.

## Convenciones

Los enunciados de los problemas especifican tipos de dato con nombres genéricos `void`, `int`, `int64`, y `int[]`(array).

En cada uno de los lenguajes de programación aceptados, los calificadores utilizan tipos de dato o implementaciones apropiadamente como se listan abajo:

Lenguaje	<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>	longitud del array <code>a</code>
C++	<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector&lt;int&gt;</code>	<code>a.size()</code>
Java	<code>void</code>	<code>int</code>	<code>long</code>	<code>int[]</code>	<code>a.length</code>

## Límites

Problema	Tiempo límite	Memoria límite
line	solo-salida	solo-salida
vision	1 segundo	1024 MB
walk	4 segundos	1024 MB