



# Notas

Si no puedes enviar tu solución mediante CMS, haz lo siguiente:

- Pon tu solución en un directorio llamado `submit_<nombre del problema>` en el escritorio (el directorio ya existe) **Antes que termine el concurso**.
  - Para el problema de solo salida los archivos se deben llamar `01.out`, `02.out`, ..., `10.out`.
  - Para los demás problemas debe estar **solo un** archivo en el folder conteniendo el código de tu solución.
- Pídele a tu líder que mande una apelación.

Si crees que se te debe dar más tiempo, haz lo siguiente:

- Envía una clarificación (desde el CMS o en papel) tan pronto como sea posible.
- No te levantes de tu lugar ni hables con otros competidores cuando termine el concurso.

---

Existe un paquete adjunto para cada problema que puedes descargar desde el sistema del concurso.

Para el problema de solo salida

- El paquete contiene los casos y entradas de ejemplo.
- Puedes subir muchos archivos dentro de un zip. Los archivos en el zip se deben llamar `?.out`, donde `??` es el número de caso (ej, `03.out`).
- Puedes zipear múltiples archivos usando el siguiente comando `zip output.zip *.out`
- Puedes hacer hasta 100 envíos para este problema. En cada envío puedes mandar la solución de cualquier subconjunto de los casos de prueba.

Para los demás problemas:

- El paquete contiene evaluadores de prueba, implementaciones de ejemplo, casos de prueba.
- Puedes hacer hasta 50 envíos de cada problema, debes enviar exactamente un archivo en cada envío.
- El nombre del archivo que debes enviar es el nombre de la descripción de cada tarea. Debe implementar los procedimientos descritos en la descripción utilizando los nombres de funciones tal y como se muestran en las implementaciones de

ejemplo.

- Puedes implementar funciones adicionales.
- Tus envíos no deben leer de la entrada estándar, escribir a la salida estándar o interactuar con cualquier otro archivo. Solo tienes permitido escribir a STDERR.
- Tu código no debe llamar funciones del sistema, **En particular `exit()` o `System.exit()`.**
- Cuando estés probando tu código con el evaluador de prueba, la entrada debe coincidir con el formato y límites de la descripción, para evitar compartamientos inesperados.
- En las entradas del evaluador de prueba, se usa un espacio para separar parametros en una linea, a menos que otro formato haya sido explícitamente especificado.
- Te recomendamos usar los scripts de compilación cuando pruebes tu código localmente. De lo contrario asegurate de agregar la opción `-std=gnu++14` al compilar en C++.

## Convenciones

Las descripciones de las tareas usan nombres genericos para los tipos de datos: `int`, `int64`, `int[]` (arreglo), e `int[][]` (matriz).

En cada lenguaje soportado, el evaluador usa el tipo de dato apropiado de acuerdo a la siguiente tabla:

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

Una matriz es un arreglo no vacío de arreglos de la misma longitud.

Lenguaje	<code>int[][]</code>	# de filas en la matriz <code>a</code>	#de columnas en la matriz <code>a</code>
C++	<code>std::vector&lt;std::vector&lt;int&gt;&gt;</code>	<code>a.size()</code>	<code>a[0].size()</code>
Java	<code>int[][]</code>	<code>a.length</code>	<code>a[0].length</code>

## Límites

<b>Tarea</b>	<b>Tiempo</b>	<b>Memoria</b>
linea	Solo salida	Solo salida
vision	1 segundo	1024 MB
caminata	4 segundos	1024 MB