



## Aviso

Para todas las tareas:

- Los límites están disponibles en la página "Overview" en el sistema del concurso.
- Hay un paquete de archivos adjuntos que puedes descargar del sistema del concurso, que contiene graders de ejemplo, implementaciones de ejemplo, casos de prueba de ejemplo y scripts de compilación y ejecución.
- Puedes hacer hasta 50 presentaciones para cada tarea, y tienes que presentar exactamente un archivo en cada presentación.
- Cuando pruebe sus programas con el grader de ejemplos, su entrada debe coincidir con el formato y las restricciones del enunciado de la tarea, de lo contrario, pueden producirse comportamientos no especificados.
- En las entradas del grader de ejemplo, cada dos tokens consecutivos en una línea están separados por un solo espacio, a menos que se especifique explícitamente otro formato.
- Cuando pruebe su código en su máquina local, le recomendamos que utilice los scripts de los paquetes adjuntos. Tenga en cuenta que utilizamos la opción del compilador -std=gnu++17.
- Si no puedes enviar tu código al CMS, puedes utilizar la herramienta ioisubmit para almacenarlo y evaluarlo una vez finalizado el concurso.
  - Ejecute `ioisubmit <nombre_de_tarea> <archivo_de_origen>` en el directorio con `<archivo_de_origen>`.
  - Pida a un miembro del comité que tome una foto de la salida de `ioisubmit`. Su presentación no se considerará si no se ha realizado este paso.
    - Si está compitiendo en línea, pida a su supervisor que tome una foto de la salida de `ioisubmit` y la envíe a los organizadores.

## Convención

Las sentencias de las tareas especifican firmas utilizando nombres de tipos genéricos `void`, `bool`, `int`, `int[]` (array), y `union(bool, int[])`.

En C++, los graders utilizan tipos de datos o implementaciones apropiadas, como se indica a continuación

<code>void</code>	<code>bool</code>	<code>int</code>	<code>int[]</code>
<code>void</code>	<code>bool</code>	<code>int</code>	<code>std::vector&lt;int&gt;</code>

<code>union(bool, int[])</code>	tamaño del array a
<code>std::variant&lt;bool, std::vector&lt;int&gt;&gt;</code>	<code>a.size()</code>

En C++, `std::variant` se define en el encabezado `<variant>`. Un método con el tipo de retorno `std::variant<bool, std::vector<int>>` puede devolver un `bool` o un `std::vector<int>`. El código de ejemplo que sigue muestra tres ejemplos de funciones que devuelven `std::variant`.

```
std::variant<bool, std::vector<int>> foo(int N) {
    return N % 2 == 0;
}
std::variant<bool, std::vector<int>> goo(int N) {
    return std::vector<int>(N, 0);
}
std::variant<bool, std::vector<int>> hoo(int N) {
    if (N % 2 == 0) {
        return false;
    }
    return std::vector<int>(N, 0);
}
```