



## Note

Per tutti i problemi:

- I limiti sono visibili nella pagina "Overview" del sistema di gara.
- Puoi fare fino a 50 sottoposizioni di esattamente un file.
- Sono allegati grader e implementazione di esempio, casi di esempio e script di compilazione.
- Consigliamo di usare lo script per compilare con le flag di gara, come `-std=gnu++17`.
- Usare il grader di esempio con input che non rispetta i limiti nel testo è *undefined behaviour*.
- I token in input su una stessa riga sono sempre separati da un singolo spazio.
- Se CMS non fosse disponibile, esegui `ioisubmit` per far valutare codice dopo la gara:
  - `ioisubmit <task_shortname> <source_file>` (nella cartella di `<source_file>`).
  - Chiama un membro di comitato per fotografare l'output di `ioisubmit` (**necessario**).

## Convenzioni

I testi usano i nomi di tipo generici `void`, `bool`, `int`, `int[]` (array), e `union(bool, int[])`.

In C++, i grader usano i tipi concreti:

<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>	lunghezza di a
<code>std::variant&lt;bool, std::vector&lt;int&gt;&gt;</code>	<code>a.size()</code>

In C++, `std::variant` è definito nell'header `<variant>`. Una funzione che restituisce `std::variant<bool, std::vector<int>>` può eseguire l'istruzione `return` sia con un `bool` sia con un `std::vector<int>`. Per esempio:

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