



# Upute za natjecatelje

U svim zadacima:

- Ograničenja su vidljiva na "Overview" stranici sustava za evaluaciju.
- Postoji zapakirani privitak kojeg možete preuzeti sa sustava za evaluaciju. Taj privitak sadrži ogledne ocjenjivače, ogledne implementacije, probne primjere i skripte za kompilaciju.
- Rješenja možete poslati najviše 50 puta na sustav za evaluaciju. Potrebno je poslati jednu datoteku.
- Kada testirate vaša rješenja oglednim ocjenjivačem, format i ograničenja ulaznih podataka navedena u tekstu zadatka trebaju biti ispoštovana. U protivnom, moglo bi doći do nedefiranog ponašanja (engl. *Undefined Behaviour*)
- U ulazima oglednog ocjenjivača, svaka dva uzastopna tokena u istom retku trebaju biti odvojena jednim znakom razmaka, osim ako je suprotno navedeno u tekstu zadatka.
- Kada testirate vaša rješenja na lokalno stroju, preporučamo vam da koristite skripte za kompilaciju iz zapakiranog privitka. Primijetite da koristimo `-std=gnu++17` opciju prilikom kompilacije.
- Ako ne možete poslati rješenje na CMS, možete koristiti alat `ioisubmit` da biste spremili vaš kod za evaluaciju nakon kraja natjecanja.
  - Pokrenite `ioisubmit <task_shortname> <source_file>` u direktoriju gdje se nalazi `<source_file>`.
  - Zamolite vašeg proktora (Fabijan, Paula, Ivan, Gospodin) da uslikaju izlaz alata `ioisubmit` i pošalju ga organizatorima. Slanje rješenja (engl. *submission*) neće biti valjano bez ovog koraka.

## Konvencije

Tekstovi zadataka specificiraju potpise funkcija koristeći generička imena tipova `void`, `int`, `int[]` (polje) i `union(bool, int[])`.

U programskom jeziku C++, ocjenjivači koriste odgovarajuće tipove podataka ili implementacije prema sljedećoj tablici:

<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>
<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector&lt;int&gt;</code>

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

U programskom jeziku C++, `std::variant` je definiran u zaglavlju `<variant>`. Metoda povratnog tipa `std::variant<bool, std::vector<int>>` može vratiti rezultat ili tipa `bool` ili tipa `std::vector<int>`. Sljedeći ogledni kod prikazuje tri ispravna primjera funkcije koje kao povratnu vrijednost vraćaju `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);
}
```