



Opombe

Za vse naloge:

- Splošne omejitve lahko najdete v sistemu za tekmovanje na strani "Overview".
- Paket prilog, ki si ga lahko prenesete iz sistema za tekmovanje, vsebuje vzorčne ocenjevalnike, vzorčne rešitve, osnovne testne primere in skripte za prevajanje in izvajanje.
- Za vsako nalogo lahko naredite največ 50 oddaj, vsaka oddaja mora vsebovati natanko eno samo datoteko.
- Ko svoje programe preverjate z vzorčnim ocenjevalnikom, poskrbite, da bo vaš vhod zadoščal predpisanimi obliki in omejitvam iz opisa naloge, sicer lahko pride do neželenega obnašanja programa.
- Pri vseh vzorčnih ocenjevalnikih sta dva zaporedna žetona v vrstici ločena z enojnim presledkom razen, če ni izrecno zahtevana drugačna oblika.
- Za testiranje vaše kode na lokalnem računalniku priporočamo uporabo skript iz paketa prilog. Upoštevajte tudi, da se pri prevajanju uporablja stikalo `-std=gnu++17`.
- Če imate težave z oddajo kode na CMS, lahko uporabite orodje `ioisubmit`, ki omogoča pripravo kode za naknadno oceno po koncu tekmovanja.
 - Izvedite na način: `ioisubmit <task_shortname> <izvorna_datoteka>` v mapi, kjer se nahaja `<izvorna_datoteka>`.
 - Člana odbora prosite, da naredi posnetek izhoda programa `ioisubmit`. V primeru, da ta korak ne bo uspešno izveden, vaša oddaja ne bo upoštevana.
 - Če tekmujete oddaljeno, prosite svojega skrbnika, da naredi posnetek izhoda programa `ioisubmit` in jo pošlje organizatorjem tekmovanja.

Konvencije

Opisi nalog uporabljajo imena generičnih podatkovnih tipov `void`, `bool`, `int`, `int[]` (array), in `union(bool, int[])`.

Pri programskem jeziku C++ vzorčni ocenjevalniki uporabljajo ustrezne podatkovne tipe ali implementacije, kot so navedene spodaj

<code>void</code>	<code>bool</code>	<code>int</code>	<code>int[]</code>
<code>void</code>	<code>bool</code>	<code>int</code>	<code>std::vector<int></code>

<code>union(bool, int[])</code>	dolžina polja a
<code>std::variant<bool, std::vector<int>></code>	<code>a.size()</code>

V programskem jeziku C++ je `std::variant` definiran v glavi z imenom `<variant>`. Metoda, ki vrača vrednost podatkovnega tipa `std::variant<bool, std::vector<int>>`, lahko vrača bodisi `bool` bodisi `std::vector<int>`. Programska koda prikazana spodaj, prikazuje tri delujoče primere funkcij, ki vračajo `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);
}
```