



## Vispārīga informācija

Visiem uzdevumiem:

- Ierobežojumi ir pieejami testēšanas sistēmas lapā "Overview".
- Ir pielikumu pakotne, kuru varat lejupielādēt no testēšanas sistēmas. Tajā atrodas paraugvērtētāji, paraugimplementācijas, testpiemēri, kā arī kompilēšanas un palaišanas skripti.
- Katram uzdevumam varat veikt līdz 50 iesūtījumiem, kur katrā iesūtījumā jābūt tieši vienam failam.
- Pārbaudot programmas ar paraugvērtētāju, jūsu ievaddatiem ir jāatbilst uzdevumā minētajam formātam un ierobežojumiem, pretējā gadījumā izpildes rezultāts nav definēts.
- Paraugvērtētāja ievadā katri divi secīgi vienumi rindā tiek atdalīti ar vienu tukšumzīmi, ja vien skaidri nav norādīts cits formāts.
- Pārbaudot kodu savā datorā, mēs iesakām izmantot pielikumu pakotnēs dotos skriptus. Lūdzu, ņemiet vērā, ka tiks izmantota kompilatora opcija `-std=gnu++17`.
- Ja nevarat iesniegt CMS, varat izmantot rīku "ioisubmit", lai saglabātu kodu novērtēšanai pēc sacensību beigām.
  - Palaidiet `ioisubmit <uzd_īsaīs_nosaukums> <programmas_teksta_fails>` direktorijā, kurā atrodas `<programmas_teksta_fails>`.
  - Palūdziet kādam no komitejas locekļiem nofotografēt 'ioisubmit' rezultātu. Jūsu iesniegums netiks izskatīts, ja šī darbība nebūs veikta.

## Vienošanās

Uzdevumu tekstos tiek izmantotas signatūras, lietojot vispārīgus tipu nosaukumus `void`, `bool`, `int`, `int[]` (masīvs), `union(bool, int[])`.

Valodā C++, vērtētāji izmanto atbilstošus datu tipus vai implementācijas, kā parādīts zemāk:

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

Valodā C++, `std::variant` ir definēts `<variant>` hederī. Metode ar atgriežamās vērtības tipu `std::variant<bool, std::vector<int>>` var atgriezt vai nu `bool`, vai `std::vector<int>`. Tālāk dotajā koda fragmentā parādīti trīs darbojošies piemēri, kur izmantotas funkcijas ar atgriežamo tipu `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);
}
```