



Pastabos

Visoms užduotims:

- Laiko ir atminties ribojimai matomi varžybų sistemos *Overview* puslapyje.
- Iš varžybų sistemos galite parsisiųsti failų paketą, kuriame yra pavyzdinės vertinimo programos, pavyzdinės realizacijos, pavyzdiniai testai ir kompiliavimo bei paleidimo komandos.
- Kiekvienai užduočiai sprendimą galite pateikti ne daugiau nei 50 kartų, kiekvieną sprendimą sudaro lygiai vienas failas.
- Testuodami savo sprendimą su pavyzdine vertinimo programa įsitikinkite, kad jūsų pateikiami duomenys atitinka formatą ir ribojimus, pateiktus užduoties aprašyme; priešingu atveju gali būti neapibrėžta elgsena.
- Pavyzdinių vertinimo programų duomenyse eilutėje gretimi skaičiai ar žodžiai atskiriami vienu tarpu, nebent aiškiai nurodytas kitoks formatas.
- Testavimui savo kompiuteriuose rekomenduojame naudoti failų paketuose esančias kompiliavimo komandas. Atkreipkite dėmesį, kad šios komandos naudoja `-std=gnu++17` kompiliavimo nuostatą.
- Jei nepavyksta pateikti sprendimo per CMS, galite naudoti `ioisubmit` įrankį, kuris išsaugos jūsų sprendimą, kad jis būtų įvertintas pasibaigus varžyboms.
 - Paleiskite `ioisubmit <užduotis> <sprendimo_failas>` kataloge, kuriame yra `<sprendimo_failas>`.
 - Paprašykite komiteto nario nufotografuoti `ioisubmit` rezultatą. Jei to nepadarysite, jūsų sprendimas nebus vertinamas.

Sutartiniai žymėjimai

Užduočių aprašymuose funkcijos aprašomos naudojant bendrinius tipų vardus `void`, `int`, `int64`, `int[]` (masyvas) ir `union(bool, int[])`.

C++ kalboje vertinimo programos naudoja atitinkamus duomenų tipus, pateiktus lentelėje:

<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>	masyvo a ilgis
<code>std::variant<bool, std::vector<int>></code>	<code>a.size()</code>

C++ kalboje `std::variant` yra apibrėžtas `<variant>` antraštėje (header). Funkcija, kurios grąžinamos reikšmės tipas yra `std::variant<bool, std::vector<int>>`, gali grąžinti arba `bool` arba `std::vector<int>`. Žemiau esantis pavyzdinis kodas parodo tris funkcijų, grąžinančių `std::variant`, veikiančius pavyzdžius.

```
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);
}
```