



Precizări

Pentru toate problemele:

- Limitele sunt disponibile în pagina "Overview" a interfeței de concurs.
- Există o arhivă atașată pe care o puteți descărca de pe interfața de concurs, care conține gradere locale, exemple de teste, precum și scripturi de compilare și rulare.
- Puteți face până la 50 de submisii la fiecare problemă și trebuie să încărcați un singur fișier la fiecare submisie.
- Când testați programul cu grader-ul local, intrarea trebuie să corespundă formatului solicitat și să respecte restricțiile din enunț, în caz contrar pot apărea erori neprecizate.
- Pentru datele de intrare ale grader-ului, oricare două valori consecutive de pe aceeași linie sunt separate printr-un singur spațiu, cu excepția cazului când se specifică altfel.
- Când vă testați codul pe mașina locală recomandăm să folosiți scripturile din arhiva atașată. Rețineți că folosim opțiunea de compilare: `-std=gnu++17`.
- Dacă nu puteți submita pe CMS, folosiți utilitarul `ioisubmit` pentru a stoca submisia în vederea evaluării după terminarea concursului.
 - Rulați comanda `ioisubmit <task_shortname> <source_file>` în folderul în care se află `<source_file>`.
 - Solicitați unui membru al comisiei să facă o poză a rezultatului comenzii `ioisubmit`. Submisia va fi luată în considerare doar dacă acest pas a fost efectuat.
 - Dacă concurați online, cereți supraveghetorului să facă o poză a rezultatului comenzii `ioisubmit` și să o trimită organizatorilor.

Convenție

Enunțurile problemelor specifică prototipurile funcțiilor folosind nume generice de tipuri: `void`, `bool`, `int`, `int[]` (tablou unidimensional), și `union(bool, int[])`.

În C++, gradele folosesc tipuri de date sau implementări adecvate, după cum este enumerat mai jos

<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>	length of array a
<code>std::variant<bool, std::vector<int>></code>	<code>a.size()</code>

În C++ `std::variant` este definit în headerul `<variant>`. O metodă definită ca returnând tipul `std::variant<bool, std::vector<int>>` poate returna fie `bool`, fie `std::vector<int>`.

Mai jos sunt date trei exemple de lucru cu funcții care returnează `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);
}
```