



Бележки

За всички задачи:

- Ограниченията са достъпни в страницата "Общ преглед" в системата на състезанието.
- Има пакет с прикачени файлове, който можете да изтеглите от състезателната система, съдържащ примерни грейдъри, примерни реализации, примерни тестови случаи и скриптове за компилация и изпълнение.
- Можете да направите до 50 изпращания за всяка задача и трябва да изпратите точно един файл във всяко изпращане.
- Когато тествате вашите програми с примерния грейдер, вашият вход трябва да съответства на формата и ограниченията от условието на задачата; в противен случай може да възникне неопределено поведение.
- В примерните входове на грейдъра всеки два последователни токена на ред са разделени с един интервал, освен ако изрично не е указан друг формат.
- Когато тествате кода си на вашата локална машина, ви препоръчваме да използвате скриптове от пакетите с прикачени файлове. Моля, имайте предвид, че ние използваме опцията за компилатор -std=gnu++17.
- Ако не можете да изпратите в CMS, можете да използвате инструмента `ioisubmit`, за да съхраните кода си за оценка след края на състезанието.
 - Изпълнете `ioisubmit <task_shortname> <source_file>` в директория с `<source_file>`.
 - Помолете член на комисията да направи снимка на резултата от „ioisubmit“. Вашето изпращане няма да бъде разгледано, освен ако тази стъпка не е направена.
 - Ако се състезавате онлайн, помолете вашия проктор да направи снимка на резултата от „ioisubmit“ и да я изпрати на организаторите.

Конвенции

В условията на задачите се използват означения чрез следните имена на общи типове `void`, `int`, `int64`, `int[]` (масив), и `int[][]` (масив от масиви).

В C++ грейдърът използва подходящи типове данни или реализации, както е изброено по-долу

<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<int></code>

<code>int[][]</code>	дължина на масива <code>a</code>
<code>std::vector<std::vector<int>></code>	<code>a.size()</code>

В C++ `std::variant` е дефиниран в хедъра `<variant>`. Метод, който връща стойност от типа `std::variant<bool, std::vector<int>>`, може да върне `bool` или `std::vector<int>`. Примерният код по-долу показва три работещи примера на функции, връщащи `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);
}
```