



Not

Tüm görevler için:

- Limitler (sınırlar) yarışma sistemindeki "Overview(Genel Bakış)" sayfasında mevcuttur.
- Yarışma sisteminden indirebileceğiniz, örnek değerlendiriciler, örnek programlar, örnek test senaryoları, derleme ve çalıştırma scriptlerini içeren bir paket bulunmaktadır.
- Her görev için 50'ye kadar gönderim yapabilirsiniz ve her gönderimde tam olarak bir dosya göndermeniz gerekir.
- Programlarınızı örnek değerlendirici ile test ederken, girdiniz görev tanımındaki format ve sınırlamalar ile eşleşmelidir, aksi takdirde belirtilmemiş davranışlar oluşabilir.
- Örnek değerlendirici girdilerinde, başka bir format açıkça belirtilmedikçe, bir satırdaki her iki ardışık girdi tek bir boşlukla ayrılır.
- Kodunuzu yerel makinenizde test ettiğinizde eklenmiş paketlerdeki scriptleri kullanmanızı öneririz. Lütfen -std=gnu++17 derleyici seçeneğini kullandığımızı unutmayın.
- CMS'ye gönderemiyorsanız, yarışmanın bitiminden sonra kodunuzu değerlendirme için ioisubmit aracını kullanabilirsiniz.
- `<source_file>` içeren dizinde `ioisubmit <task_shortname> <source_file>` komutunu çalıştırın.
- Bir komite üyesinden ioisubmit çıktısının fotoğrafını çekmesini isteyin. Bu adım yapılmadıkça gönderiminiz dikkate alınmayacaktır.
- Çevrimiçi yarışıyorsanız, gözetmeninizden ioisubmit çıktısının fotoğrafını çekmesini ve organizatörlere göndermesini isteyin.

Kural

Görev tanımları, `void`, `bool`, `int`, `int[]` (dizi), ve `union(bool, int[])`. genel tür adlarını kullanarak imzaları (signature) belirtir

C++'da, aşağıda listelendiği gibi değerlendiriciler uygun veri türlerini ve kodlamaları kullanır.

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

C++'da `std::variant` , `<variant>` header'ı içinde tanımlanmıştır.

Return türü `std::variant<bool, std::vector<int>>` olan metot `bool` ya da `std::vector<int>` return edebilir.

Aşağıdaki örnek kod `std::variant` return eden fonksiyon için üç tane örnek gösterir.

```
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);
}
```