



نکات مهم

برای تمام سوالات:

- محدودیت‌ها در بخش "Overview" بر روی cms قابل مشاهده هستند.
- برای هر سوال یک attachment package وجود دارد که می‌توانید از روی cms دانلود کنید، این پکیج شامل sample implementations, example test cases و sample graders, compile and run scripts است.
- برای هر سوال حداکثر ۵۰ سبمیت می‌توانید انجام دهید و در هر سبمیت می‌توانید دقیقاً یک فایل را سبمیت کنید.
- وقتی برنامه را با گریدر داده شده تست می‌کنید دقت کنید که فرمت ورودی شما دقیقاً مطابق با فرمت گفته شده در صورت سوال باشد، در غیر اینصورت رفتار نامشخصی (unspecified behaviors) ممکن است رخ دهد.
- در ورودی sample grader، هر دو توکن متوالی توسط یک space جدا شده‌اند، مگر آنکه فرمت دیگری به وضوح در صورت سوال گفته شده باشد.
- وقتی برنامه را روی سیستم خود تست می‌کنید، توصیه می‌شود که همان اسکریپتی که در attachment package هست را استفاده کنید. توجه کنید که جاج از `std=gnu++17` استفاده می‌کند.
- اگر سبمیت در cms ممکن نبود، می‌توانید از دستور `ioisubmit` استفاده کنید تا کد شما برای سبمیت ذخیره شود.
 - در فولدري که کدتان را ذخیره کرده‌اید دستور `ioisubmit <task_shortname>`
 - `<source_file>` را در ترمینال وارد کنید. (به جای `task_shortname` نام کوتاه سوال و به جای `source_file` نام فایلی که کدتان در آن موجود است را وارد کنید)
 - حتماً بعد از این کار از یکی از اعضای کمیته بخواهید که از خروجی این دستور عکس بگیرد، در غیر اینصورت این سبمیت در نظر گرفته نمی‌شود.

قرارداد

در صورت سوالات از تایپ‌های عمومی نام برده شده است، مانند: `void, int, int64, int[], int[][]`.

در C++ معادل آن‌ها به صورت زیر است:

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

در C++، تایپ `std::variant` در هدر `<variant>` تعریف شده است.

یک تابع که خروجی آن تایپ `std::variant<bool, std::vector<int>>` را دارد می‌تواند `bool` یا `std::vector<int>` برگرداند. در کدهای نمونه‌ی زیر ۳ مثال از توابعی که `std::variant` به عنوان خروجی بر می‌گرداند آمده است.

```
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);
}
```