ملاحظات

من أجل جميع المسائل:

- تحدد الحدود limits ضمن صفحة Overview في نظام المسابقة
- هنالك حزمة مرفقة يمكنك تنزيلها من نظام المسابقة، تحتوي الحزمة على محكّم sample grader، وتحقيق sample implementation، وأمثلة example test cases، وسكريبتات الترجمة والتشغيل compile and run scripts.
- يمكنك ارسال 50 ارسالية على كل مسألة، حيث عليك ارسال ملف واحد فقط في كل ارسالية.
- عند اختبار برنامجك باستخدام المحكّم sample grader، يجب أن يطابق الدخل تنسيق وقيود المسألة. عدا عن ذلك، يمكن أن ينتج سلوك غير معروف.
- في دخل المحكّم sample grader، يفصل بين كل رمزين tokens متتاليين على سطر واحد مسافة واحدة فقط، إلا اذا تم التصريح بغير ذلك.
- عند اختبار برنامجك على حاسبك المحلي، ننصح باستخدام السكريبتات الموجودة في الحزمة المرفقة. لاحظ استخدام `-std=gnu++17` في سكريبت الترجمة.
- إذا لم تتمكن من الارسال على CMS لسبب ما، يمكنك استخدام الأداة `ioisubmit` لحفظ الكود للتحكيم بعد انتهاء المسابقة.
  - شغل `ioisubmit <task_shortname> <source_file>` في المجلد الذي يحتوي الكود `<source_file>`.
  - اطلب أحد أعضاء اللجنة التقاط صورة لخرج الأداة `ioisubmit`. لن يتم اعتبار الارسالية الخاصة بك ما لم تتبع هذه الخطوة.
    - إذا كنت مشاركاً عن بعد online، اطلب من المراقب التقاط الصورة لخرج الأداة `ioisubmit` وارسالها إلى اللجنة التنظيمية.

العرف

The task statements specify signatures using generic type names `void`, `int`, `int64`, `int[]` (array), and `int[][]` (array of array).

In C++, the graders use appropriate data types or implementations, as listed below

void	**bool**	**int**	**int[]**
`void`	`bool`	`int`	`std::vector<int>`

union(bool, int[])	**length of array a**
`std::variant<bool, std::vector<int>>`	`a.size()`

In C++, `std::variant` is defined in the `<variant>` header. A method with the return type `std::variant<bool, std::vector<int>>` can return either a `bool` or an `std::vector<int>`. The sample code below shows three working examples of functions returning `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);
}
```