



ประกาศ

สำหรับโจทย์แต่ละข้อ

- ข้อจำกัดจะอยู่ในหน้า "Overview" ของระบบการแข่งขัน
- ชุดไฟล์แนบ สามารถดาวน์โหลดจากระบบการแข่งขัน ประกอบด้วยเกรดเดอร์ตัวอย่าง ตัวอย่างการเขียนโปรแกรม ชุดทดสอบตัวอย่าง และสคริปต์คอมไพล์และรัน
- แต่ละข้อสามารถส่งได้ไม่เกิน 50 ครั้ง และในการส่ง (submission) แต่ละครั้ง ต้องส่งหนึ่งไฟล์เท่านั้น
- เมื่อทดสอบโปรแกรมของนักเรียนด้วยเกรดเดอร์ตัวอย่าง ข้อมูลที่ป้อนควรเข้ากันกับรูปแบบและข้อจำกัดจากคำสั่ง มิฉะนั้นอาจเกิดพฤติกรรมที่ไม่พึงประสงค์
- ในข้อมูลนำเข้าของเกรดเดอร์ตัวอย่าง ถ้ามีข้อมูลมากกว่าหนึ่งตัวในหนึ่งบรรทัด ข้อมูลที่ติดกันจะถูกคั่นด้วยช่องว่าง (space) 1 ช่อง ยกเว้นกรณีที่โจทย์นั้นระบุรูปแบบเฉพาะที่ต่างไป
- เมื่อนักเรียนทดสอบโค้ดบนเครื่องของนักเรียน ขอแนะนำให้นักเรียนใช้สคริปต์ในชุดไฟล์แนบ โปรดทราบว่าขอให้ใช้ออปชันคอมไพเลอร์ `-std=gnu++17`
- หากเกิดเหตุที่ทำให้นักเรียนไม่สามารถ submit ที่ CMS นักเรียนสามารถใช้เครื่องมือ `ioisubmit` เพื่อส่งโค้ดของนักเรียนเข้าที่จัดเก็บ เพื่อรอการตรวจให้คะแนนหลังจบการแข่งขัน
 - รัน `ioisubmit <task_shortcode> <source_file>` ในไดเรกทอรีที่มี `<source_file>`
 - ให้กรรมการคุมสอบถ่ายภาพผลลัพธ์ของ `ioisubmit` หากไม่ปฏิบัติตามขั้นตอนนี้จะไม่ตรวจให้
 - ถ้าสอบแบบออนไลน์ ให้กรรมการคุมสอบถ่ายรูปผลลัพธ์ `ioisubmit` และส่งให้ผู้จัดงานด้วย

ข้อตกลง

ในโจทย์แต่ละข้อจะกำหนดชนิดของข้อมูลเป็น `void` , `int` , `int64` , `int[]` (อาร์เรย์), และ `int[][]` (อาร์เรย์ 2 มิติ) การเขียนโปรแกรมภาษา C++ ตัวเกรดเดอร์กำหนดให้ใช้ชนิดข้อมูล ตามรูปแบบต่อไปนี้

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

ใน C++ จะมีการกำหนด `std::variant` ในส่วนหัว `<variant>`

เมธอดชนิดส่งคืนค่า `std::variant<bool, std::vector<int>>` สามารถคืนค่า `bool` หรือ `std::vector<int>` ก็ได้

โค้ดตัวอย่างด้านล่างเป็นการแสดง 3 ตัวอย่างการทำงานของฟังก์ชันคืนค่า `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);
}
```