



## הודעה

בכל המשימות:

- המגבלות זמינות בעמוד "Overview" במערכת התחרות.
- יש חבילת קבצים מצורפים שאתם יכולים להוריד ממערכת התחרות, המכילה גריידרים לדוגמה, דוגמאות מימוש, טסטים לדוגמה וסקריפטים לקומפילציה ולהרצה.
- אתם יכולים לבצע עד 50 הגשות בכל שאלה, ועליכם להגיש בדיוק קובץ אחד בכל הגשה.
- בעת בדיקת התוכנית שלכם עם הגריידר לדוגמה, על הקלט שלכם להתאים לפורמט ולמגבלות מתיאור השאלה. אחרת, התנהגויות לא מוגדרות עלולות להתרחש.
- בקלטים לגריידר לדוגמה, כל שני סימבולים עוקבים בשורה מופרדים ע"י רווח בודד, אלא אם כן פורמט אחר צוין במפורש.
- כאשר אתם בודקים את הקוד שלכם לוקאלית, אנו ממליצים לכם להשתמש בסקריפטים שבחבילת הקבצים המצורפים. שימו לב שאנו משתמשים בדגל `std=gnu++17` - של הקומפיילר.
- אם אינכם מסוגלים להגיש ל-CMS, אתם יכולים להשתמש בכלי `ioisubmit` כדי לשמור את הקוד שלכם לבדיקה לאחר סוף התחרות.
- הריצו `<source_file> <task_shortname> ioisubmit` בתיקייה המכילה את `<source_file>`.
- בקשו מחבר וועדה לצלם תמונה של הפלט של `ioisubmit`. ההגשה שלכם לא תיחשב אם צעד זה לא בוצע.
- אם אתם מתחרים באופן מקוון, בקשו מהמשיגי שלכם לצלם תמונה של הפלט של `ioisubmit` ולשלוח אותה למארגנים.

## קונבנציות

בתיאור השאלות משתמשים בחתימות עם שמות הטיפוסים הגנריים `int`, `bool`, `int[]` (מערך), ו-`union(bool, int[])`.

ב-C++, הגריידרים משתמשים בטיפוסים או במימושים המתאימים, כמתואר מטה

<code>void</code>	<code>bool</code>	<code>int</code>	<code>int[]</code>
<code>void</code>	<code>bool</code>	<code>int</code>	<code>std::vector&lt;int&gt;</code>

האורך של המערך <code>a</code>	<code>union(bool, int[])</code>
<code>a.size()</code>	<code>std::variant&lt;bool, std::vector&lt;int&gt;&gt;</code>

ב-C++, הטיפוס `std::variant` מוגדר בקובץ ה-`<variant>`. פונקציה עם ערך חזרה מסוג `std::variant<bool, std::vector<int>>` יכולה להחזיר או `bool` או `std::vector<int>`. הקוד לדוגמה מטה מציג שלוש דוגמאות תקינות לפונקציות המחזירות `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);
}
```