



# Notizen

Falls das Einsenden über CMS nicht klappen sollte:

- Speichere deine Lösungsdateien **vor Ende des Contests** auf dem Desktop in einem Ordner `submit_<task name>` ab. (Dieser Ordner sollte bereits existieren.)
- Bei "Output-Only Problemen" musst du die Dateien mit `01.out`, `02.out`, ..., `10.out` benennen.
- Bei anderen Problemen muss **genau ein** Datei in diesem Ordner sein, welches deinen Quellcode enthält.
- Dein Team-Leader muss einen Einspruch erheben.

Wenn du glaubst, dass du zusätzliche Zeit bekommen müsstest (bei technischen Problemen), gehe bitte folgendermassen vor.

- Schicke so bald als möglich einen "clarification request" (über das Contestsystem oder auf Papier)
- Verlasse deinen Arbeitsplatz nicht und sprich auch nicht mit anderen Teilnehmern nach dem Ende des Wettbewerbes.

Jede Aufgabe hat einen Anhang, der von dir vom Wettbewerbsystem herunter geladen werden kann. Es gibt "Output-Only" Aufgaben für die gilt:

- Der Anhang enthält Input-Testfälle und Beispiel-Testfälle. Jeder Testfall ist ein separater Subtask.
- Du kannst mehrere Ausgabedateien als ZIP-Archiv schicken. Zu diesem Zweck müssen die Ausgabedatei mit `?.out` benannt werden, wobei `??` die Testfallnummer ist (zum Beispiel: `03.out`).

Du kannst mehrere Dateien mit dem Kommando `zip output.zip *.out` zippen.

- Du darfst bis zu 100 Einsendungen bei Output-Only Aufgaben schicken. Bei jeder Einsendung darfst du eine beliebige Teilmenge von Testfällen schicken.

Für alle Aufgaben gilt:

- Du kannst dir ein ZIP-Archiv vom CMS herunterladen, welches Beispiel-Grader, Musterdateien, einige Testfälle und Kompilierungsskripte enthält.
- Du darfst jede Aufgabe bis zu 50-mal einsenden. Jede Einsendung besteht aus genau einer Datei.
- Der Name der Datei, die du einsenden solltest, steht in der Kopfzeile der

Aufgabenstellung. Alle in der Aufgabenstellung angegebenen Funktionen müssen implementiert werden. Die genauen Signaturen der Funktionen sind den Musterdateien zu entnehmen.

- Es steht dir frei weitere Funktionen zu implementieren.
- Deine Einsendungen dürfen nicht **exit()** or **System.exit()** aufrufen.
- Wenn du dein Programm mit dem Beispiel-Grader testest, muss dein Input das Format und die Beschränkungen der Aufgabenstellung respektieren, sonst kann es zu unerwartetem Verhalten kommen.
- In den Beispiel-Grader-Inputs sind aufeinanderfolgende Tokens der gleichen Zeile durch ein einziges Leerzeichen getrennt, solange kein anderes Format angegeben ist.
- Wenn du dein Programm lokal testest, empfehlen wir dir die Skripte aus dem ZIP-Archiv zu benutzen. Solltest du das nicht tun, dann stelle sicher, dass C++-Programme mit der Option `-std=gnu++14` kompiliert werden.

## Konventionen

Die Aufgabenstellungen benutzen generische Typen wie `void`, `int`, `int64` und `int[]` (Array).

Die Grader verwenden, je nach Sprache, die generischen Typen folgendermaßen:

Sprache	<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>	Länge von Array <code>a</code>
C++	<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector&lt;int&gt;</code>	<code>a.size()</code>
Java	<code>void</code>	<code>int</code>	<code>long</code>	<code>int[]</code>	<code>a.length</code>

## Beschränkungen

Aufgabe	Zeitlimit	Speicherlimit
line	output-only	output-only
vision	1 sec	1024 MB
walk	4 sec	1024 MB