



Nie ľad (Notice)

Ak pri konci súťaže neviete odovzdať riešenia cez CMS, postupujte nasledovne:

- Súbory, ktoré chcete odovzdať, uložte do adresára `submit_<task name>` na desktope. (Tento adresár by už mal existovať.) Toto musíte stihnúť spraviť **pred koncom súťaže**.
 - Pre úlohu, ktorá je output-only, sa súbory v tomto adresári musia volať `01.out`, `02.out`, ..., `10.out`.
 - Pre ostatné typy úloh musíte do adresára uložiť **práve jeden** súbor a ten musí obsahovať zdrojový kód, ktorý chcete odovzdať.
- Po súťaži požiadajte svojho team leadera, aby spravil appeal.

Ak si myslíte, že si (z ľubovoľného dôvodu) zaslúžite **extra čas** na riešenie úloh po skončení súťaže:

- Čo najskôr to oznámte to organizátorom, a to tak, že buď cez contest system alebo na papieri pošlete clarification request.
- Po skončení súťaže ostaňte sedieť pri svojom stole a nerozprávajte sa s ostatnými súťažiacimi.

Ku každej úlohe si zo systému môžete stiahnuť balíček.

Niektoré úlohy sú output-only. Pre tie platí:

- Balíček obsahuje súťažné vstupy a príklady vstupu a výstupu. Každý súťažný vstup predstavuje samostatnú podúlohu.
- Keď odovzdávate riešenia tejto úlohy, môžete ho odovzdať ako zip súbor obsahujúci vaše výstupy k viacerým súťažným vstupom. Tieto súbory sa musia volať `?.out`, kde `??` je číslo príslušného vstupu (napr. `03.out`). Keď máte v adresári viacero takto pomenovaných súborov, zip súbor s nimi vyrobíte nasledovným príkazom: `zip output.zip *.out`
- Pre output-only úlohu môžete spraviť až 100 submitov. V každom submite môžete odovzdať ľubovoľnú podmnožinu výstupov.

Pre ostatné úlohy platí:

- V balíčku sú ukážkové gradery, ukážkové implementácie, príklady vstupu a výstupu, a skripty na kompiláciu.
- Riešenie každej úlohy môžete odovzdať nanajvýš 50-krát, pričom vždy odovzdávate práve jeden súbor.

- Názov súboru, ktorý odovzdávate, je v hlavičke zadania. Majú v ňom byť implementované všetky funkcie, ktoré zadanie úlohy vyžaduje. Tieto funkcie musia mať deklaráciu (presnejšie signatúru) rovnakú ako tie v ukázkovej implementácii.
- Môžete implementovať vlastné pomocné funkcie.
- Nesmiete nič sami čítať zo štandardného vstupu, vypisovať na štandardný výstup, ani čítať a písať z/do súborov.
- Smiete vypisovať na štandardný chybový výstup (stderr).
- Vaše riešenie **nesmie volať** funkciu `exit()`, resp. `System.exit()`
- Pri testovaní svojich programov pomocou ukázkového gradera dajte pozor, aby ste dodržali jeho formát vstupu a obmedzenia zo zadania. Ak tak nespravíte, je správanie ukázkového gradera nedefinované.
- Vo vstupoch pre ukázkový grader sú tokeny v riadku od seba oddelené práve jednou medzerou, ak nie je v zadaní povedané inak.
- Pri testovaní vašich programov na lokálnom počítači je odporúčané použiť kompilačný skript z balíčka. Ak tak nespravíte a programujete v C++, ubezpečte sa aspoň, že medzi prepínačmi kompilátora máte aj `-std=gnu++14`.

Konvencie

V zadaniach sú použité generické typy `void`, `int`, `int64` a `int[]` (pole).

V podporovaných programovacích jazykoch týmto generickým typom zodpovedajú nasledovné konkrétne typy:

Jazyk	<code>void</code>	<code>int</code>	<code>int64</code>	<code>int[]</code>	dĺžka poľa <code>a</code>
C++	<code>void</code>	<code>int</code>	<code>long long</code>	<code>std::vector<int></code>	<code>a.size()</code>
Java	<code>void</code>	<code>int</code>	<code>long</code>	<code>int[]</code>	<code>a.length</code>

Limity

Úloha	Časový limit	Pamäťový limit
line	nemá	nemá
vision	1 s	1024 MB
walk	4 s	1024 MB