

## Egy kis kellemetlenség (light)

A Tudományos Bizottság már pihen az idei CEOI megnyitó ünnepségén. A feladatok elkészültek, az értékelő szerver mind a  $10^{12}$  tűzfala végre működik és a bizottság élvezné a műsort a lángoló fáklyákkal. Semmi sem mehet rosszul. Kivéve... senki sem vett elég olajat a fáklyákhoz! A bizottságnak segítségére van szüksége, hogy a műsort úgy tudják lebonyolítani, hogy ne fogyjon el az olajkészletük.

A műsor alatt a fellépők egy sorban állnak, balról jobbra, 1-től kezdve sorszámozva. Az előadók száma időről-időre változik. Minden előadónál van egy fáklya, amely lehet, hogy ég, de az is lehet, hogy nem. Kezdetben csak egyetlen előadó van a színpadon, akinek a fáklyája ég.

Az előadás  $Q$  felvonás-ból áll. Az  $a$ . felvonás kezdetén vagy  $p_a > 0$  előadó csatlakozik a sorhoz a jobb oldalon, vagy a jobb szélső  $p_a > 0$  előadó távozik a sorból. A csatlakozók/a sort elhagyók számát a bizottság nem tudja befolyásolni. Azt tudjuk, hogy a baloldali előadó (az egyes sorszámú) mindig a színpadon marad. A csatlakozó előadók fáklyái nem égnek, a távozó előadók eloltják a fáklyáikat, ha azok égtek.

Amikor az  $a$ . felvonás előadói már sorbaálltak, a bizottságnak kell mondania egy  $t_a \geq 0$  számot. Ezután minden előadó, akinek a fáklyája ég, megosztja a tüzet a tőle jobbra lévő  $t_a$  előadóval. Ez azt jelenti, hogy az  $i$ . előadó fáklyája akkor és csak akkor fog égni, ha legalább egy előadó fáklyája, akinek a sorszáma  $\max\{i - t_a, 1\}, \dots, i$  már korábban is égett. *A dinamikus előadás érdekében  $t_a$  nem lehet nagyobb, mint  $5p_a$ , és a lehető legkisebbnek kell lennie* (lásd az alábbi, pontozásról szóló részt).

Az egyes felvonások végén a bizottság minden előadónak, akinek a fáklyája éppen ég, megmondja, hogy el kell-e azt oltania vagy sem. Esztétikai okokból a jobb szélső előadó fáklyájának égve kell maradnia. *Az olaj spórolása érdekében az égve hagyott fáklyák száma nem lehet nagyobb, mint 150.* Írj egy programot, amely segít a bizottságnak, hogyan döntsön az egyes felvonásoknál, a fenti feltételekkel.

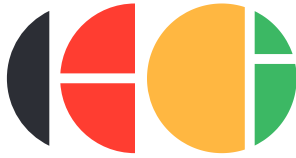
## Interakció

Ez egy interaktív feladat. A következő három függvényt kell megvalósítanod:

- ▶ **void prepare()** a futtatás elején hívódik meg. Ezt a függvényt használhatod a beállításokhoz (vagy hagyhatod úgy, hogy nem csinál semmit).
- ▶ **pair<long long, vector<long long>> join(long long  $p_a$ )** akkor hívódik meg, amikor a  $p_a > 0$  előadó csatlakozik a sor jobb oldalához. Ennek a függvénynek egy olyan párt kell visszaadnia, amelynek első tagja a bizottság által megadott  $t_a$  számot tartalmazza és a második része azoknak az előadóknak a listájából áll, akiknek a fáklyájának a felvonás végén égve kell maradnia. *Ez a lista szigorúan növekvő sorrendben kell, hogy legyen.*
- ▶ **pair<long long, vector<long long>> leave(long long  $p_a$ )** akkor hívódik meg, amikor a jobb szélső  $p_a > 0$  előadó elhagyja a sort. A függvénynek egy olyan párt kell visszaadnia, amely a bizottság által megadott  $t_a$  számból és azoknak az előadóknak a listájából áll, akiknek a műsorszám végén égve kell maradnia a fáklyájának (ezek sorszámainak, szigorúan növekvő sorrendben).

Ha a visszatérési értékek bármelyike nem felel meg a fenti feltételeknek, a program azonnal megszakad, és az adott tesztesetben **Not correct** értéket kap. Nem írhat sz a standard kimenetre és nem olvashatsz a standard bemenetről; ellenkező esetben a **Security violation!** értékelést kaphatod. Azonban a szabványos hibafolyamba (error stream) szabadon írhat sz (`stderr`).

A `light.h` fájlt be kell építened (include) a forráskódodba. A program helyi teszteléséhez a `sample_grader.cpp` állományt használhatod, amely a cms-ben a feladathoz tartozó mellékletek közt található. A mintatesztelő



használatának leírását lásd alább, és a `sample_grader.cpp`-ben találsz segítséget arra vonatkozóan is, hogyan futtasd a programoddal. A melléklet tartalmaz egy minta megvalósítást is, `light_sample.cpp` néven.

## Határok és értékelés

Jelölje  $N$  az egyidejűleg sorban álló előadók maximális számát. Minden tesztesetben,  $N \leq 10^{17}$  and  $1 \leq Q \leq 50\,000$ .

**Subtask 1 (5 points).** A `leave()` függvénynek csak egy hívása van tesztesetenként.

**Subtask 2 (5 points).**  $N \leq 700$

**Subtask 3 (10 points).**  $N \leq 5\,000$

**Subtask 4 (5 points).**  $N \leq 25\,000$

**Subtask 5 (10 points).**  $N \leq 100\,000$

**Subtask 6 (5 points).**  $N \leq 500\,000$

**Subtask 7 (60 points).** Nincs további megkötés.

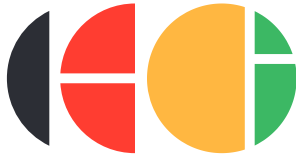
**Részpontozás.** A 7. részfeladatban a tényleges pontszám a  $t_a/p_a$ -az összes  $a$  felvonásra vonatkozó - maximális értékétől függ, az alábbi táblázat szerint:

$\max t_a/p_a$	[0, 1]	(1, 2]	(2, 3]	(3, 5]
pont	60	35	20	10

A teljes pontszámhoz a `join()` és a `leave()` függvény összes visszatérési értékére igaz kell legyen, hogy  $t_a \leq p_a$ .

## Minta interakció

Tekintsünk egy  $Q = 4$  értékű tesztesetet. Itt a program és az értékelő közötti interakció a következőképpen nézhet ki:



Call	Return value	Explanation
<code>prepare()</code>	—	Itt bármilyen beállítást elvégezhetesz (vagy nem csinálsz semmit). Az előadás egy előadóval kezdődik, akinek a fáklyája ég. 
<code>join(3)</code>	3, {2, 4}	Három előadó csatlakozik, így összesen négy előadó lesz a sorban. Az 1. előadó meggyújtja a 2., a 3. és a 4. előadó fáklyáit; ezt követően az 1. és a 3. előadó eloltja a fáklyáját. 
<code>leave(2)</code>	0, {2}	A két jobb szélső előadó távozik (csak ketten maradnak). Nem gyújtanak új fáklyát; a 2. előadó fáklyája égve marad. 
<code>join(2)</code>	3, {2, 4}	Két előadó csatlakozik, így összesen négy előadó lesz a sorban. A 2. előadó meggyújtja a 3. és a 4. előadó fáklyáját; a 3. előadó ezt követően eloltja a fáklyáját. 
<code>join(3)</code>	3, {2, 4, 7}	Három előadó csatlakozik, így összesen hét előadó lesz a sorban. Először a 3., az 5., a 6. és a 7. előadó fáklyái gyulladnak meg, majd a 3., az 5. és a 6. előadók ismét eloltják fáklyáikat. 

A fenti interakciót a `light_sample.cpp` futtatásakor jelenik meg, a nyilvános tesztesetben.  
Megjegyzés: a `join()` és a `leave()` hívások fenti sorozata érvényes teszteset bármelyik részfeladatban.

## Értékelő

A mintaértékelő először a standard bemeneten a  $Q$  egész számot várja. Ezután a standard kimenetre írja a fenti három függvény összes hívását.

Az elején az értékelő a `prepare()` parancsot hívja. Ezután szimulálja a  $Q$  felvonást. Minden egyes  $a$  felvonásnál a standard bemeneten vagy egy  $p_a > 0$  egész számot vár, ami azt jelenti, hogy  $p_a$  előadó csatlakozik a sorhoz, vagy egy  $q_a < 0$  egész számot, ami azt jelenti, hogy  $p_a := -q_a$  előadó elhagyja a sort; ekkor meghívja a `join( $p_a$ )`, illetve a `leave( $p_a$ )` parancsot.

Befejezéskor a következő üzenetek egyikét írja a standard kimenetre:

**Invalid input.** Az értékelő a standard bemeneten keresztül nem a fenti formátumban kapta az adatokat.

**The stage is empty.** Kevesebb, mint egy előadó maradt a sorban.

**Invalid return value.** A  $t_a$  visszatérési érték nem felel meg a  $0 \leq t_a \leq 5p_a$  feltételnek, az előadók listája nem szigorúan növekvő sorrendben lett megadva, vagy az előadók listája 1-nél kisebb vagy a színpadon lévő előadók létszámánál nagyobb egész számot tartalmaz.

**Too many burning torches.** Több mint 150 fáklya égett a hívás után.



# CEOI 2023

Central-European Olympiad in Informatics  
Magdeburg | Germany | August 13 - 19

**Day 1**  
**Task: light**  
**Language: en**

**Rightmost torch not on fire.** A jobb oldali fáklya nem égett, vagy kialudt.

**Not all announced torches have been lit.** A függvényed által visszaküldött előadók közül legalább egynek a fáklyája nem égett.

**Correct: ratio at most  $f$ , at most  $b$  burning torches.** A fenti hibák egyike sem fordult elő, a `leave()` és a `join()` minden hívása kielégítette a  $t_a \leq f \cdot p_a$  feltételt (kerekítési hibáig), és legfeljebb  $b$  fáklya égett bármelyik hívás után.

Ezzel szemben a ténylegesen, a CMS-ben használt értékelő csak **Not correct** (a fenti hibák bármelyikére), a **Security violation!**, a **Partially correct** vagy a **Correct** értékelést adja. Az értékelő *adaptív*, azaz az egyes felvonásokban belépő vagy kilépő előadók száma függhet a programod viselkedésétől (az aktuális és a korábbi hívásoktól is). Mind a mintaértékelő, mind a programod elbírálására használt értékelő automatikusan megszakítja a programodat, ha a fenti hibák valamelyike előfordul.

## Limits

Time: 1 s

Memory: 512 MiB