

Mince, váha a BOLívia

V rámci získavania praktických životných skúseností sa Olívia rozhodla opustiť Kult Skrytého Poznania a začala pracovať v MicroStepe. V práci sa jej darí, dokonca zarobila svojich prvých šesť mincí - a tiež vďaka nej získala titul bakalára. Andy Lúčny je však veľmi svojrázny pán a preto to nie sú obyčajné mince. Tieto mince sú očíslované od **1** po **6** a každá minca má inú váhu. A za reálne peniaze jej ich Andy vymení iba vtedy, ak ich správne usporiada od najľahšej po najťažšiu.

Na to by Olívia rada použila Bašinu dokonale presnú kuchynskú váhu. Baša v tom ale hneď spoznala skvelú príležitosť ako Olíviu týrať a odmietla jej ju požičať. Je však Olívii ochotná odpovedať na špeciálne položené otázky.

Baša postavila pred Olíviu štyri krabičky označené **A**, **B**, **C** a **D**. Po vložení niektorých mincí do krabičiek sa môže Olívia spýtať jednu zo štyroch možných otázok. Pri každom type otázky sa v každej krabičke **A**, **B** a **C** musí nachádzať **práve** jedna minca a v prípade štvrtej otázky musí byť navyše aj **práve** jedna minca v krabičke **D**. (Pri ostatných typoch otázky je krabička **D** prázdna.)

Otázky, ktoré sa môže Olívia Bašky pýtať sú nasledovné:

1. Ktorá z mincí v krabičkách **A**, **B** a **C** je najťažšia?
2. Ktorá z mincí v krabičkách **A**, **B** a **C** je najľahšia?
3. Ktorá z mincí v krabičkách **A**, **B** a **C** je medián? (To je tá z nich, ktorá nie je ani najľahšia ani najťažšia.)
4. Medzi mincami v krabičkách **A**, **B** a **C** uvažujme iba tie, ktoré sú ťažšie ako minca v krabičke **D**. Ktorá z týchto mincí je najľahšia? V prípade, že všetky mince sú ľahšie ako minca v **D**, Baša ako odpoveď uvedie úplne najľahšiu spomedzi mincí z krabičiek **A**, **B** a **C**.

Úloha

Napíšte program, ktorý zoradí Olíviine mince od najľahšej po najťažšiu. Váš program môže klásť Baši otázky tak ako boli definované vyššie. Každý vstup bude obsahovať viacero test casov, pričom každý test case bude zodpovedať inej množine mincí.

Váš program musí implementovať funkcie `init()` a `orderCoins()`. Pri každom spustení programu grader najskôr raz zavolá funkciu `init()`. Z parametra funkcie `init()` sa dozviete počet test casov v tomto vstupe a môžete inicializovať hodnoty ľubovoľných premenných vo vašom programe. Grader potom niekoľkokrát zavolá funkciu `orderCoins()`: postupne raz pre každý test case v tomto vstupe.

- `init(T)`
 - `T`: počet test casov, ktoré obsahuje tento vstup. Platí, že $1 \leq T \leq 18$.
 - Táto funkcia nemá žiadnu návratovú hodnotu.
- `orderCoins()`

- Táto funkcia je zavolaná práve raz pre každý test case vstupu.
- Táto funkcia by mala zistiť správne poradie Olíviiných mincí pomocou volaní funkcií `getHeaviest()`, `getLightest()`, `getMedian()` a `getNextLightest()`.
- Keď funkcia zistí správne poradie mincí, mala by to oznámiť zavolaním funkcie `answer()`.
- Po zavolaní funkcie `answer()`, musí funkcia `orderCoins()` skončiť bez návratovej hodnoty.

Vo svojom programe môžete použiť nasledujúce funkcie, ktoré sú implementované v rámci gradera.

- `answer(W)` — pomocou tejto funkcie odovzdáva váš program nájdené poradie.
 - `W`: pole dĺžky **6** obsahujúce mince zoradené v správnom poradí. Hodnoty `W[0]` až `W[5]` musia obsahovať čísla mincí (čísla od **1** po **6**) v poradí od najľahšej po najťažšiu mincu.
 - Táto funkcia má byť zavolaná iba z funkcie `orderCoins()` a to práve raz pre každý test case.
 - Táto funkcia nemá návratovú hodnotu.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — tieto funkcie postupne zodpovedajú otázkam číslo **1**, **2** a **3**.
 - `A, B, C`: čísla mincí, ktoré budú umiestnené do krabičiek **A**, **B** a **C**. Musia to byť tri rôzne čísla z rozsahu **1** až **6**.
 - Každá z týchto funkcií vráti jednu z hodnôt `A, B` a `C` — číslo mince, ktorá je odpoveďou na danú otázku. Napríklad `getHeaviest(A, B, C)` vráti číslo najťažšej z troch zadaných mincí.
- `getNextLightest(A, B, C, D)` — táto funkcia zodpovedá otázke číslo **4**.
 - `A, B, C, D`: čísla mincí, ktoré budú umiestnené do krabičiek **A**, **B**, **C** a **D**. Musia to byť štyri rôzne čísla z rozsahu **1** až **6**.
 - Funkcia vráti jednu z hodnôt `A, B` a `C` — číslo mince, ktorá je odpoveďou na danú otázku. Teda číslo tej mince, ktorá je najľahšia spomedzi mincí v krabičkách **A**, **B** a **C**, ktoré sú ťažšie ako minca v krabičke **D**; alebo, v prípade že sú všetky ľahšie ako minca v krabičke **D**, číslo najľahšej mince spomedzi mincí v krabičkách **A**, **B** a **C**.

Bodovanie

Táto úloha neobsahuje žiadne podúlohy. Namiesto toho bude vaše riešenie bodované podľa počtu otázok (teda zavolaní funkcií `getHeaviest()`, `getLightest()`, `getMedian()` a `getNextLightest()`), ktoré váš program spraví.

Váš program bude spustený na viacerých vstupoch, každý vstup bude obsahovať viacero test casov. Nech r je počet vstupov, na ktorých bude spustený váš program (toto číslo je fixné). Ak váš program neutriedi mince správne, na ľubovoľnom test case v ľubovoľnom vstupe, dostanete za celú úlohu **0** bodov. V opačnom prípade budete obodovaní zvlášť pre každý vstup, a to nasledovným spôsobom.

Nech Q je najmenšie také číslo, že je možné usporiadať ľubovoľnú permutáciu šiestich mincí použitím

najviac Q otázok. Samozrejme, hodnotu Q vám neprezradíme.

Predpokladajte, že najväčší počet otázok, ktoré položí váš program na niektorom test case v ľubovoľnom vstupe je $Q + y$ pre nejaké kladné číslo y . Pozrime sa teraz na jeden vstup vyriešený vašim programom. Nech $Q + x$ (pre nezáporné celé x) je najväčší počet otázok, ktorý použije váš program v rámci jedného test casu tohoto vstupu. (Ak na každom test case použijete menej ako Q otázok, tak $x = 0$.) Skóre pre tento vstup bude $\frac{100}{r((x+y)/5+1)}$, zaokrúhlené **nadol** na dve desatinné miesta.

Ak váš program položí najviac Q otázok v každom test case každého vstupu, dostanete **100** bodov.

Príklad

Predpokladajme, že mince sú zoradené od najľahšej po najťažšiu v poradí **3 4 6 2 1 5**.

Volaná funkcia	Návratová hodnota	Vysvetlenie
getMedian(4, 5, 6)	6	Minca číslo 6 je medián spomedzi mincí 4, 5 a 6 .
getHeaviest(3, 1, 2)	1	Minca číslo 1 je spomedzi mincí 1, 2 a 3 najťažšia.
getNextLightest(2, 3, 4, 5)	3	Mince 2, 3 a 4 sú ľahšie ako minca 5 , takže výsledkom je najľahšia z nich (minca 3).
getNextLightest(1, 6, 3, 4)	6	Mince 1 a 6 sú ťažšie ako minca 4 . Spomedzi týchto mincí je ľahšia minca 6 .
getHeaviest(3, 5, 6)	5	Minca 5 je najťažšia spomedzi mincí 3, 5 a 6 .
getMedian(1, 5, 6)	1	Minca 1 je medián spomedzi mincí 1, 5 a 6 .
getMedian(2, 4, 6)	6	Minca 6 je medián spomedzi mincí 2, 4 a 6 .
answer([3, 4, 6, 2, 1, 5])		Program zistil správnu odpoveď pre daný test case.

Sample grader

Sample grader číta vstup v nasledujúcom formáte:

- prvý riadok: číslo T — počet test casov
- druhý až $T + 1$ -vý riadok — permutácia čísel **1 až 6** udávajúca poradie mincí od najľahšej po najťažšiu.

Napríklad, vstup pozostávajúci z dvoch test casov, s poradím mincí **1 2 3 4 5 6** a **3 4 6 2 1 5** vyzerá nasledovne:

```
2
1 2 3 4 5 6
3 4 6 2 1 5
```

Sample grader vypíše pole, ktoré dostal ako parameter funkcie `answer()`.