



Vaga

Amina ima šest novčića, označenih brojevima od **1** do **6**. Ona zna da svi novčići imaju različitu težinu, po kojoj bi ih htjela sortirati. Za tu priliku osmislila je novu vrstu vage.

Tradicionalna vaga ima dvije posude. Koristi se tako da se u svaku posudu stavi jedan novčić i vaga odredi koji je novčić teži.

Aminina nova vaga je složenija. Ona ima četiri posude, označene s **A**, **B**, **C** i **D**. Vaga ima četiri različite postavke, od kojih svaka odgovora na različito pitanje. Da bi koristila vagu Amina mora staviti točno jedan novčić u svaku od posuda **A**, **B** i **C**. Dodatno, kod četvrte postavke također mora staviti jedan novčić u posudu **D**.

Četiri postavke odgovaraju sljedećim pitanjima:

1. Koji od novčića u posudama **A**, **B** i **C** je najteži?
2. Koji od novčića u posudama **A**, **B** i **C** je najlakši?
3. Koji od novčića u posudama **A**, **B** i **C** je srednji po težini?
4. Među novčićima u posudama **A**, **B** i **C**, promatrajmo samo one novčiće koji su teži od novčića u posudi **D**. Ukoliko postoji barem jedan takav, koji od njih je najlakši? U suprotnom, ukoliko nema takvih novčića, koji od novčića u posudama **A**, **B** i **C** je najlakši?

Zadatak

Napišite program koji će sortirati Amininih šest novčića po težini. Program može davati upite Amininoj vagi kako bi usporedio težine novčića. Vašem programu će biti zadano da riješi nekoliko test primjera, pri čemu svakom odgovara novi skup novčića.

Vaš program mora implementirati funkcije `init` i `orderCoins`. Pri svakom izvršavanju vašeg programa, grader će prvo pozvati `init` točno jednom. To će vam dati broj test primjera i mogućnost da inicijalizirate neke varijable. Grader će nakon toga pozivati `orderCoins()`, po jednom za svaki test primjer.

- `init(T)`
 - `T`: Broj test primjera koji vaš program mora riješiti prilikom ovog izvršavanja. `T` je cijeli broj iz intervala **1, ..., 18**.
 - Ova funkcija nema povratnu vrijednost.
- `orderCoins()`
 - Ova funkcija se poziva točno jednom za svaki test primjer.
 - Ova funkcija treba odrediti točan redoslijed Amininih novčića pozivajući funkcije gradera

`getHeaviest()`, `getLightest()`, `getMedian()` i/ili `getNextLightest()`.

- Jednom kada ova funkcija odredi koji je točan poredak, treba ga javiti pozivajući funkciju `gradera answer()`.
- Nakon pozivanja `answer()`, funkcija `orderCoins()` mora završiti. Ova funkcija nema povratnu vrijednost.

U vašem programu možete koristiti sljedeće funkcije `gradera`:

- `answer(W)` — vaš program koristi ovu funkciju da javi redoslijed koji je pronašao.
 - W : Niz dužine 6 koji sadrži točan poredak novčića. $W[0]$ do $W[5]$ trebaju biti oznake novčića (dakle, brojevi od **1** do **6**) u poretku od najlakšeg do najtežeg novčića.
 - Ovu funkciju vaš program smije da pozivati samo iz `orderCoins()`, jednom za svaki test primjer.
 - Ova funkcija nema povratnu vrijednost.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — ove funkcije odgovaraju redom postavkama 1, 2 i 3 Aminine vage.
 - A, B, C : Novčići koji su redom stavljeni u posude **A**, **B** i **C**. A, B i C moraju biti tri različita cijela broja, svaki između **1** i **6** uključivo.
 - Svaka funkcija vraća jedan od brojeva A, B i C : oznaku odgovarajućeg novčića. Na primjer `getHeaviest(A, B, C)` vraća oznaku najtežeg od tri zadana novčića.
- `getNextLightest(A, B, C, D)` — ovo odgovara postavci 4 Aminine vage.
 - A, B, C, D : Novčići redom stavljeni u posude **A**, **B**, **C** i **D**. A, B, C i D moraju biti četiri različita broja, svaki između **1** i **6** uključivo.
 - Funkcija vraća jedan od brojeva A, B i C : oznaku novčića koji je vaga dala kao odgovor pri postavci 4. Tj. vraćeni novčić je najlakši od onih novčića u posudama **A**, **B** i **C** koji su teži od novčića u posudi **D**; ili, ukoliko nijedan od njih nije teži od novčića u posudi **D**, vraćeni novčić je najlakši od tri novčića u posudama **A**, **B**, and **C**.

Bodovanje

U ovom zadatku ne postoje podzadaci. Umjesto toga vaš broj bodova će biti baziran na broju vaganja (ukupan broj poziva funkcija `gradera` `getLightest()`, `getHeaviest()`, `getMedian()` i/ili `getNextLightest()`) koje vaš program napravi.

Vaš program će biti izvršavan više puta sa više test primjera u svakom izvršavanju. Neka je r broj izvršavanja vašeg programa. Ovaj broj je fiksiran u test podacima. Ako vaš program ne nađe ispravan poredak novčića u bilo kojem test primjeru bilo kojeg izvršavanja, dobit ćete 0 bodova. U suprotnom, svako izvršavanje se boduje pojedinačno na sljedeći način.

Neka je Q najmanji broj takav da je moguće sortirati bilo koji niz šest novčića sa Q vaganja na Amininoj vagi. Kako bi vam zadatak učinili težim, nećemo vam ovdje otkriti vrijednost broja Q .

Predpostavimo da je najveći broj vaganja među svima test primjerima u svim izvršavanjima jednak $Q + y$ za neki cijeli broj y . Sada promatrajmo jedno izvršavanje vašeg programa. Neka je najveći broj

vaganja u svih T test primjera u tom izvršavanju jednak $Q + x$ za neki nenegativan cijeli broj x . (Ako koristite manje od Q vaganja u svakom test primjeru, onda je $x = 0$.) Konačno, broj bodova za ovo izvršavanje iznosi $\frac{100}{r((x+y)/5+1)}$, zaokružen *na dolje* na dvije decimale.

Na primjer, ako vaš program napravi najviše Q vaganja u svakom test primjeru svakog izvršavanja, dobit ćete 100 bodova.

Primjer

Pretpostavimo da je sortirani niz novčića **3 4 6 2 1 5** od najlakšeg do najtežeg.

poziv funkcije	povratna vrijednost	objašnjenje
getMedian(4, 5, 6)	6	Novčić 6 je srednji po težini među novčićima 4 , 5 i 6 .
getHeaviest(3, 1, 2)	1	Novčić 1 je najteži među novčićima 1 , 2 i 3 .
getNextLightest(2, 3, 4, 5)	3	Novčići 2 , 3 i 4 su svi lakši od novčića 5 , pa je najlakši među njima (3) vraćen.
getNextLightest(1, 6, 3, 4)	6	Novčići 1 i 6 su oba teža od novčića 4 . Među novčićima 1 i 6 , novčić 6 je najlakši.
getHeaviest(3, 5, 6)	5	Novčić 5 je najteži među novčićima 3 , 5 i 6 .
getMedian(1, 5, 6)	1	Novčić 1 je srednji po težini među novčićima 1 , 5 i 6 .
getMedian(2, 4, 6)	6	Novčić 6 je srednji po težini među novčićima 2 , 4 i 6 .
answer([3, 4, 6, 2, 1, 5])		Program je pronašao točan odgovor za ovaj test primjer.

Lokalni grader

Grader čita ulazne podatke u sljedećem formatu:

- redak **1**: T — broj test primjera
- svaki od redaka **2** do $T + 1$: niz od **6** različitih brojeva od **1** do **6**: poredak novčića od najlakšeg do najtežeg.

Na primjer, ulaz koji se sastoji od dva test primjera u kojima su novčići u porecima **1 2 3 4 5 6** i **3 4 6 2 1 5** izgleda ovako:

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

Grader ispisuje niz koji je zadan kao argument funkciji `answer()`.