



Filetova magična vaga

File Piroćanac ima šest novčića, označenih od **1** do **6**. On zna da nikoja dva novčića nemaju istu težinu. Želeo bi da ih sortira prema njihovoj težini. Za tu priliku osmislio je novu vrstu vage.

Tradicionalna vaga ima dva tase. Koristi se tako što se postavi novčić na svaki tas i vaga odredi koji je novčić teži.

Filetova nova vaga je komplikovanija. Ona ima četiri tase, označenih sa **A**, **B**, **C** i **D**. Vaga ima četiri različita podešavanja, od kojih svako odgovora na različito pitanje u vezi sa novčićima. Da bi koristio vagu File mora postaviti tačno jedan novčić na svaki od tasova **A**, **B** i **C**. Dodatno, pri četvrtom podešavanju takođe mora postaviti jedan novčić na tas **D**.

Četiri podešavanja daju naredbu vagi da odgovori na sledeća pitanja:

1. Koji od novčića na tasovima **A**, **B** i **C** je najteži?
2. Koji od novčića na tasovima **A**, **B** i **C** je najlakši?
3. Koji od novčića na tasovima **A**, **B** i **C** je srednji po težini?
4. Među novčićima na tasovima **A**, **B** i **C**, posmatrajmo samo one novčiće koji su teži od novčića na tasu **D**. Ukoliko postoji bar jedan takav, koji od njih je najlakši? U suprotnom, ukoliko ne postoji takvih novčića, koji od novčića na tasovima **A**, **B** i **C** je najlakši?

Zadatak

Napisati program koji će sortirati Filetovih šest novčića prema njihovoj težini. Program može zadavati upite Filetovoj vagi da bi poredio težine novčića. Vašem programu će biti dato da reši nekoliko test primera, pri čemu svakom odgovara novi skup novčića.

Vaš program treba da implementira funkcije `init` i `orderCoins`. Pri svakom izvršavanju vašeg programa, grejder će prvo pozvati `init` tačno jednom. To će vam dati broj test primera i mogućnost da inicijalizujete neke promenljive. Grejder će nakon toga pozivati `orderCoins()`, po jednom za svaki test primer.

- `init(T)`
 - `T`: Broj test primera koji vaš program treba da reši prilikom ovog izvršavanja. `T` je ceo broj iz intervala **1, ..., 18**.
 - Ova funkcija nema povratnu vrednost.
- `orderCoins()`
 - Ova funkcija se poziva tačno jednom za svaki test primer.
 - Ova funkcija treba da odredi tačan redosled Filetovih novčića pozivajući funkcije grejdera

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

- Jednom kada ova funkcija shvati koji je tačan redosled, treba da ga saopšti pozivajući funkciju grejdera `answer()`.
- Nakon pozivanja `answer()`, funkcija `orderCoins()` treba da se završi. Ova funkcija nema povratnu vrednost.

U vašem programu možete koristiti sledeće funkcije grejdera:

- `answer(W)` — vaš program treba da koristi ovu funkciju da saopšti odgovor koji je pronašao.
 - W : Niz dužine 6 koji sadrži tačan redosled novčića. $W[0]$ do $W[5]$ treba da budu oznake novčića (dakle, brojevi od **1** do **6**) u redosledu od najlakšeg do najtežeg novčića.
 - Ovu funkciju vaš program sme da poziva samo iz `orderCoins()`, jednom za svaki test primer.
 - Ova funkcija nema povratnu vrednost.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — ove funkcije odgovaraju podešavanjima 1, 2 i 3 Filetove vage, respektivno.
 - A, B, C : Novčići koji su stavljeni na tasove **A**, **B** i **C**, respektivno. A, B i C moraju biti tri različita cela broja, svaki između **1** i **6** uključujući i **1** i **6**.
 - Svaka funkcija vraća jedan od brojeva A, B i C : oznaku odgovarajućeg novčića. Na primer `getHeaviest(A, B, C)` vraća oznaku najtežeg od tri data novčića.
- `getNextLightest(A, B, C, D)` — ovo odgovara podešavanju 4 Filetove vage.
 - A, B, C, D : Novčići stavljeni na tasove **A**, **B**, **C** i **D**, respektivno. A, B, C i D moraju biti četiri različita broja, svaki između **1** i **6** uključujući i **1** i **6**.
 - Funkcija vraća jedan od brojeva A, B i C : oznaku novčića koji je vaga dala kao odgovor pri podešavanju 4, tj. vraćeni novčić je najlakši od onih novčića na tasovima **A**, **B** i **C** koji su teži od novčića na tasu **D**; ili, ukoliko nijedan od njih nije teži od novčića na tasu **D**, vraćeni novčić je najlakši od tri novčića na tasovima **A**, **B**, and **C**.

Bodovanje

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

Vaš program će biti izvršavan više puta sa više test primera u svakom izvršavanju. Neka je r broj izvršavanja vašeg programa. Ova broj je fiksiran u test podacima. Ako vaš program ne nađe ispravan redosled novčića u bilo kom test primeru bilo kog izvršavanja, dobićete 0 poena. U suprotnom, svako izvršavanje se boduje pojedinačno na sledeći način.

Neka je Q najmanji broj takav da je moguće sortirati bilo koji niz šest novčića sa Q vaganja na Filetovoj vagi. Da bi vas dodatno namučili, nećemo vam ovde otkriti vrednost broja Q .

Pretpostavimo da je najveći broj vaganja među svima test primerima u svim izvršavanjima jednak $Q + y$ za neki ceo broj y . Sada posmatrajmo pojedinačno izvršavanje vašeg programa. Neka je

najveći broj vaganja među svih T test primera u tom izvršavanju jednak $Q + x$ za neki nenegativan ceo broj x . Ako koristite manje od Q vaganja u svakom test primeru, onda je $x = 0$. Konačno, broj bodova za ovo izvršavanje iznosiće $\frac{100}{r((x+y)/5+1)}$, zaokružen *na dole* na dve decimale.

Na primer, ako vaš program napravi najviše Q vaganja u svakom test primeru svakog izvršavanja, dobićete 100 bodova.

Primer

Pretpostavimo da su novčići poređani u redosledu **3 4 6 2 1 5** od najlakšeg do najtežeg.

Function call	Returns	Explanation
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 tačan odgovor za ovaj test primer.

Primer grejdera

Grejder čita ulazne podatke u sledećem formatu:

- linija **1**: T — broj test primera
- linije **2, ..., T + 1**: niz od **6** različitih brojeva od **1** do **6**: redosled novčića od najlakšeg do najtežeg.

Na primer, ulaz koji se sastoji od dva test primera u kojima su novčići u redosledima **1 2 3 4 5 6** i **3 4 6 2 1 5** izgleda na sledeći način:

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

Grejder ispisuje niz koji je prosleđen kao parametar funkciji `answer()`.