



## Scales

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

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

Filetova nova vaga je komplikovanija. Ona ima četiri tasa, oznaćenih sa **A**, **B**, **C** i **D**. Vaga ima četiri različita podešavanja, od kojih svako odgovora na različito pitanje u vezi sa novćićima. Da bi koristio vagu File mora postaviti taćno jedan novćić na svaki od tasova **A**, **B** i **C**. Dodatno, pri ćetvrtom podešavanju takoće mora postaviti jedan novćić na tas **D**.

Ćetiri podešavanja daju naredbu vagi da odgovori na sljedeća pitanja:

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

## Zadatak

Napisati program koji će sortirati Filetovih šest novćića prema njihovoj težini. Program moće zadavati upite Filetovoj vagi da bi uporećivao težine novćića. Vašem programu će biti dato da riješi nekoliko test primjera, pri ćemu svakom primjeru odgovara novi skup novćića.

Vaš program treba da implementira funkcije `init` i `orderCoins`. Pri svakom izvršavanju vašeg programa, ocjenjivać (grader) će prvo pozvati `init` taćno jednom. To će vam dati broj test primera i mogućnost da inicijalizujete neke promjenljive. Ocjenjivać će nakon toga pozivati `orderCoins()`, po jednom za svaki test primjer.

- `init(T)`
  - `T`: Broj test primjera koji će vaš program trebati da riješi prilikom ovog izvršavanja. `T` je cio broj iz intervala **1, ..., 18**.
  - Ova funkcija nema povratnu vrijednost.
- `orderCoins()`
  - Ova funkcija se poziva taćno jednom za svaki test primjer.
  - Ova funkcija treba da odredi taćan redosljed Filetovih novćića pozivajuć funkcije

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

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

U vašem programu možete koristiti sljedeće funkcije ocjenjivača:

- `answer(W)` — vaš program treba da koristi ovu funkciju da saopšti odgovor koji je pronašao.
  - `W`: Niz dužine 6 koji sadrži tačan redosljed 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 primjer.
  - 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 potproblemi. Umesto toga vaš broj bodova će biti baziran na broju vaganja (ukupan broj poziva funkcija ocjenjivača `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. Ova broj je fiksiran u test podacima. Ako vaš program ne nađe ispravan redosljed novčića u bilo kom test primjeru bilo kog izvršavanja, dobićete 0 poena. 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 Filetovoj vagi. Da bi vas dodatno namučili, nećemo vam ovde otkriti vrijednost broja  $Q$ .

Pretpostavimo da je najveći broj vaganja među svima test primjerima u svim izvršavanjima jednak

$Q + y$  za neki cio broj  $y$ . Sada posmatrajmo pojedinačno izvršavanje vašeg programa. Neka je najveći broj vaganja među svih  $T$  test primjera u tom izvršavanju jednak  $Q + x$  za neki nenegativan cio broj  $x$ . (Ako koristite manje od  $Q$  vaganja u svakom test primjeru, onda je  $x = 0$ .) Konačno, broj bodova za ovo izvršavanje iznosiće  $\frac{100}{r((x+y)/5+1)}$ , zaokružen *naniže* na dvije decimale.

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

## Primjer

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

Function call	Returns	Explanation
getMedian(4, 5, 6)	6	Novčić <b>6</b> je srednji po težini među novčićima <b>4, 5 i 6</b> .
getHeaviest(3, 1, 2)	1	Novčić <b>1</b> je najteži među novčićima <b>1, 2 i 3</b> .
getNextLightest(2, 3, 4, 5)	3	Novčići <b>2, 3 i 4</b> su svi lakši od novčića <b>5</b> , pa je najlakši među njima ( <b>3</b> ) vraćen.
getNextLightest(1, 6, 3, 4)	6	Novčići <b>1 i 6</b> su oba teža od novčića <b>4</b> . Među novčićima <b>1 i 6</b> , novčić <b>6</b> je najlakši.
getHeaviest(3, 5, 6)	5	Novčić <b>5</b> je najteži među novčićima <b>3, 5 i 6</b> .
getMedian(1, 5, 6)	1	Novčić <b>1</b> je srednji po težini među novčićima <b>1, 5 i 6</b> .
getMedian(2, 4, 6)	6	Novčić <b>6</b> je srednji po težini među novčićima <b>2, 4 i 6</b> .
answer([3, 4, 6, 2, 1, 5])		Program je pronašao tačan odgovor za ovaj test primer.

## Ocjenjivač (Sample grader)

Ocjenjivač (grader) čita ulazne podatke u sljedećem formatu:

- red **1**:  $T$  — broj test primjera
- svaki od redova **2** do  $T + 1$ : niz od **6** različitih brojeva od **1** do **6**: redosljed novčića od najlakšeg do najtežeg.

Na primjer, ulaz koji se sastoji od dva test primjera u kojima su novčići u redosljedima **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
```

Ocjenjivač štampa niz koji je prosljeđen kao parametar funkciji `answer()`.