Tehtnice

Amina ima šest kovancev, označenih s števili od 1 do 6. Vsi kovanci imajo različne mase. Amina bo jih rada uredila po njihovi masi. V ta namen je razvila popolnoma novo vrsto ravnotežne tehtnice.

Tradicionalna ravnotežna tehtnica ima dve ponvici. Na vsako izmed ponvic naložimo enega izmed kovancev, tehtnica pa bo pokazala, kateri izmed kovancev je težji.

Aminina nova tehtnica je bolj zakompleksana. Ima štiri ponvice, označene s črkami A, B, C in D. Ima 4 nastavitve; pri vsaki nastavitvi odgovarja na drugačno vprašanje o kovancih. Ko Amina uporabi tehtnico, položi kovanec na vsako izmed ponvic A, B in C. Pri četrti nastavitvi pa mora položiti natanko en kovanec tudi na ponvico D.

Te štiri nastavitve bodo odgovarjale na naslednja vprašanja:

1. Kateri izmed kovancev na ponvicah A, B, in C ima največjo maso?
2. Kateri izmed kovancev na ponvicah A, B, in C ima najmanjšo maso?
3. Kateri izmed kovancev na ponvicah A, B, in C je mediana? (Kovanec, ki nima največje ali najmanjše mase med tehtanimi kovanci.)
4. Med kovanci na ponvicah A, B, in C, upoštevamo tiste kovance, ki imajo večjo maso od kovanca na ponvici D. Če taki kovanci obstajajo, kateri izmed njih ima najmanjšo maso? Če takega kovanca ni, kateri izmed kovancev na ponvicah A, B, C ima najmanjšo maso?

Naloga

Napiši program, ki bo uredil Amininih šest kovancev po njihovi masi. Program lahko kovance primerja z Aminino tehtnico. Tvojemu programu bo podanih več testnih primerov, vsak z drugimi šestimi kovanci.

Tvoj program naj vsebuje funkciji `init` in `orderCoins`. Ob vsakem zagonu programa bo na začetku funkcija `init` klicana natanko enkrat. V njej dobi program podano število testnih primerov, lahko pa nastavi tudi poljubne spremenljivke. Ocenjevalni program bo zatem klical `orderCoins()` enkrat za vsak testni primer.

- `init(T)`
  - `T`: Število testnih primerov, ki naj jih tvoj program reši. `T` je število med 1 in 18.
  - Ta metoda naj ne vrača ničesar.
- `orderCoins()`
  - Ta metoda je klicana enkrat za vsak testni primer.
  - Metoda naj ugotovi pravilen vrstni red Amininih kovancev. Kliče lahko funkcije

ocenjevalnika `getHeaviest()`, `getLightest()`, `getMedian()`, in/ali `getNextLightest()`.

- Ko metoda določi vrstni red kovancev, ga vrne tako, da kliče metodo `answer()`.
- Po klicu `answer()` naj se `orderCoins()` zaključi. Metoda naj ne vrača ničesar.

Tvoj program lahko uporablja naslednje ocenjevalnikove metode:

- `answer(W)` — tvoj program naj kliče to metodo, da vrne najdeni vrstni red kovancev.
  - `W`: Polje dolžine 6, ki vsebuje pravilen vrstni red kovancev. Elementi od `W[0]` do `W[5]` so oznake kovancev (t.j., števila med 1 in 6) razporejene od kovanca z najmanjšo do kovanca z največjo maso.
  - Tvoj program lahko to metodo kliče le iz metode `orderCoins()`, enkrat na testni primer.
  - Ta metoda naj ne vrača ničesar.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — te funkcije so enakovredne nastavitvam 1, 2 in 3 na Aminini tehtnici.
  - `A`, `B`, `C`: Kovanci v ponvicah A, B, in C, v tem zaporedju. `A`, `B`, in `C` naj bodo 3 različna cela števila med 1 in 6.
  - Vsaka izmed teh funkcij vrne eno izmed števil `A`, `B`, in `C`: število ustreznega kovanca. Na primer, `getHeaviest(A, B, C)` vrne število najtežjega izmed kovancev.
- `getNextLightest(A, B, C, D)` — ta metoda je enakovredna nastavitvi 4 na Aminini tehtnici
  - `A`, `B`, `C`, `D`: Kovanci, položeni v ponvice A, B, C, in D, tem zaporedju. `A`, `B`, `C`, in `D` naj bodo štiri različna cela števila med 1 in 6.
  - Funkcija vrne eno izmed števil `A`, `B`, in `C`: oznako kovanca, ki ga izbere tehtnica po postopku, opisanem za četrto nastavitev. To je, vrnjeni kovanec ima najmanjšo maso med kovanci na ponvicah A, B in C, ki imajo večjo maso od kovanca na ponvici D. Če takega kovanca ni, bo vrnjeni kovanec preprosto najlažji izmed kovancev na ponvicah A, B in C.

Točkovanje

Ta naloga nima podproblemov. Tvoja rešitev bo ocenjena glede na to, koliko tehtanj opravi, t.j. skupno število klicev funkcij `getLightest()`, `getHeaviest()`, `getMedian()` in/ali `getNextLightest()`.

Ocenjevani program bo pognan večkrat, z več testnimi primeri ob vsakem zagonu. Naj bo r število zagonov programa. Če program kovancev ne razporedi pravilno v katerem koli testnem primeru, dobi 0 točk. Sicer je vsak testni primer ocenjen po naslednjem postopku.

Naj bo Q najmanjše število tehtanj, s katerimi je možno urediti *poljubno* zaporedje šestih kovancev na Aminini tehtnici. Da je naloga težja, naj Q ostane naša mala skrivnost.

Naj bo $Q + y$ največje število tehtanj, ki jih je opravil ocenjevani program med vsemi testnimi primeri, med vsemi zagoni, za neko celo število y. Naj bo $Q + x$ največje število tehtanj, ki jih je ocenjevani program opravil v trenutnem testnem primeru, za neko celo število x. (Če program potrebuje manj kot

Q tehtanj za vsak testni primer, potem je $x = 0$.) Ocena programa je izračunana po naslednji formuli: $\frac{100}{r((x+y)/5+1)}$, zaokrožena *navzdol* na dve decimalki.

V splošnem velja: Če ocenjevani program opravi največ Q tehtanj v vsakem zagonu vsakega testnega primera, dobi 100 točk.

Primer

Naj bodo kovanci urejeni $3\ 4\ 6\ 2\ 1\ 5$ po naraščajoči masi.

Klic funkcije	Vrača	Razlaga
getMedian(4, 5, 6)	6	Kovanec 6 je mediana kovancev 4, 5, in 6.
getHeaviest(3, 1, 2)	1	Kovanec 1 ima največjo maso med kovanci 1, 2, in 3.
getNextLightest(2, 3, 4, 5)	3	Kovanci 2, 3, in 4 imajo vsi manjšo maso od kovanca 5, zato funkcija vrne najmanjšega med njimi (3).
getNextLightest(1, 6, 3, 4)	6	Kovanca 1 in 6 imata večjo maso od kovanca 4. Izmed kovancev 1 in 6 ima kovanec 6 manjšo maso.
getHeaviest(3, 5, 6)	5	Kovanec 5 ima največjo maso med kovanci 3, 5 in 6.
getMedian(1, 5, 6)	1	Kovanec 1 je mediana kovancev 1, 5 in 6.
getMedian(2, 4, 6)	6	Kovanec 6 je mediana kovancev 2, 4 in 6.
answer([3, 4, 6, 2, 1, 5])		Program je našel pravi odgovor za ta testni primer.

Preizkušanje

Vzorčni ocenjevalnik bere vhod v naslednji obliki:

- vrstica 1: T — število testnih primerov
- vrstice od 2 do $T + 1$: zaporedje 6 različnih števil med 1 in 6: zaporedje oznak, od tistega z najmanjšo maso, proti tistemu z največjo.

Na primer, vhod, ki sebuje 2 testna primera, v katerih so kovanci urejeni $1\ 2\ 3\ 4\ 5\ 6$ in $3\ 4\ 6\ 2\ 1\ 5$, izgleda sledeče:

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

Vzorčni ocenjevalnik izpiše polje, ki je bila podana kot parameter funkcije `answer()`.