



Svari

Aminai ir sešas monētas, kas sanumurētas ar skaitļiem no **1** līdz **6** pēc kārtas. Viņa zina, ka nekādām divām monētām nav vienāds svars. Viņa vēlas sakārtot monētas pēc to svara. Tāpēc viņa ir izgatavojusi jauna veida svarus.

Parastiem sviru svariem ir divi svaru kausi. Katrā no kausiem ieliekot pa monētai tie noteiks, kura no monētām ir smagāka.

Aminas jaunie svari ir sarežģītāki. Tiem ir četri svaru kausi, kas apzīmēti ar **A**, **B**, **C** un **D**. Svariem ir četri režīmi, un katrā no tiem iespējams iegūt atbildi uz kādu jautājumu par monētām. Lai lietotu svarus, Aminai katrā no svaru kausiem **A**, **B** un **C** jānovieto pa vienai monētai. Lietojot ceturto režīmu, viņai papildus jānovieto monēta arī svaru kausā **D**.

Četri svaru režīmi ļauj saņemt atbildes uz šādiem četriem jautājumiem:

1. Kura no kausos **A**, **B** un **C** novietotajām monētām ir smagākā?
2. Kura no kausos **A**, **B** un **C** novietotajām monētām ir vieglākā?
3. Kura no kausos **A**, **B** un **C** novietotajām monētām ir ar vidējo svaru? (Tā ir monēta, kura nav ne smagākā ne vieglākā.)
4. No kausiem **A**, **B** un **C**, ņem vērā tikai monētas, kas ir smagākas kā kausā **D** esošā monēta. Ja ir kāda tāda monēta, tad kura no tām ir vieglākā? Citādi, ja šādu monētu nav, kura no kausos **A**, **B** un **C** novietotajām monētām ir vieglākā?

Uzdevums

Uzrakstiet programmu, kas sakārto sešas Aminas monētas pēc to svara. Programma var vērsties pie Aminas svariem, lai salīdzinātu monētu svarus. Jūsu programma tiks pārbaudīta uz vairākiem testiem, kur katrā būs sešas jaunas monētas.

Jūsu programmā jāimplementē funkcijas `init` un `orderCoins`. Katrā programmas izpildes reizē vērtēšanas programma vispirms tieši vienu reizi izsauks funkciju `init`. Tas dos jums testu skaitu un ļaus inicializēt kādas mainīgo vērtības. Vērtēšanas programma pēc tam izsauks funkciju `orderCoins()` vienu reizi katram testam.

- `init(T)`
 - `T`: Testu skaits kas programmai būs jārisina šī izsaukuma laikā. `T` ir naturāls skaitlis robežās no **1** līdz **18** ieskaitot.
 - Šai funkcijai nav atgriežamās vērtības.
- `orderCoins()`
 - Šī funkcija tiek izsaukta tieši vienu reizi katram testam.

- Funkcijai ir jānoskaidro pareizā Aminas monētu secība izsaucot vērtēšanas programmas funkcijas `getHeaviest`, `getLightest`, `getMedian`, un/vai `getNextLightest`.
- Kad funkcija ir noskaidrojusi pareizo secību, tai tā ir jāpaziņo izsaucot vērtēšanas programmas funkciju `answer`.
- Pēc funkcijas `answer` izsaukuma, funkcijai `orderCoins` ir jābeidz darbs, tai nav atgriežamās vērtības.

Jūs varat lietot šadas vērtēšanas programmas funkcijas:

- `answer (W)` — jūsu programmai ir jālieto šī funkcija, lai paziņotu atrasto atbildi.
 - `W`: Sešu skaitļu masīvs ar pareizo monētu secību. `W[0]` līdz `W[5]` ir monētu numuri (skaitļi no **1** līdz **6** ieskaitot) secībā no vieglākās līdz smagākajai monētai.
 - Jūsu programmai jāizsauc šī funkcija tikai no funkcijas `orderCoins`, vienu reizi katram testam.
 - Šai funkcijai nav atgriežamās vērtības.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — šīs funkcijas atbilst Aminas svaru režīmiem 1, 2 un 3.
 - `A, B, C`: Monētu numuri, kuras novieto attiecīgi kausos **A**, **B** un **C**. `A, B` un `C` ir trīs dažādi naturāli skaitļi, katrs no **1** līdz **6** ieskaitot.
 - Katra funkcija atgriež vienu no skaitļiem `A, B` vai `C`: attiecīgās monētas numuru. Piemēram, `getHeaviest(A, B, C)` atgriezīs smagākās monētas numuru, starp šīm trim monētām.
- `getNextLightest(A, B, C, D)` — šī funkcija atbilst Aminas svaru ceturtajam režīmam.
 - `A, B, C, D`: Monētu numuri, kuras novieto attiecīgi kausos **A**, **B**, **C** un **D**. `A, B, C` un `D` ir četri dažādi naturāli skaitļi, katrs no **1** līdz **6** ieskaitot.
 - Funkcija atgriež vienu no skaitļiem `A, B` vai `C`: svaru ceturtā režīma izvēlētajās monētas numuru. Tas ir, šī monēta ir vieglākā no tām, ka atrodas kausos **A**, **B** vai **C**, kas ir smagāka kā **D** kausā novietotā monēta; vai, ja neviena no tām nav smagāka kā kausā **D** novietotā monēta, tad tā atgriež vieglāko no kausos **A**, **B** un **C** esošajām monētām.

Vērtēšana

Šajā uzdevumā nav apakšuzdevumu. Jūsu rezultātu noteiks, svēršanu skaits (kopējais funkciju `getHeaviest`, `getLightest`, `getMedian` un/vai `getNextLightest` izsaukumu skaits), ko jūsu programma izdarīs.

Jūsu programma tiks izpildīta vairākas reizes ar vairākiem testiem, katrā no tiem. Kopējais programmas izpildes reižu skaits ir r . Šis skaits ir fiksēts testēšanas datos. Ja jūsu programma nesakārto monētas pareizi kādā izpildīšanas reizes kādā testā, tā saņems 0 punktus. Citādi, katra izpildes reize tiek vērtēta atsevišķi, tālāk aprakstītajā veidā.

Ar Q apzīmēsim mazāko Aminas svaru svēršanu skaits ar kuru ir iespējams atrast sakārtojumu jebkurai sešu monētu secībai. Lai uzdevums būtu grūtāks, Q vērtība jums netiek paziņota.

Ja lielākais svēršanu skaits starp visu izpildes reižu visiem testiem ir $Q + y$, kur y ir vesels skaitlis, tad

apskatos vienu izpildes reizi, lielākais svēršanu skaits starp visiem T testiem ir $Q + x$, kur x ir nenegatīvs vesels skaitlis. (Ja jūsu programma visos testos izmanto mazāk par Q svēršanām, tad $x = 0$.) Tad šīs izpildes reizes rezultāts būs $\frac{100}{r((x+y)/5+1)}$, noapaļots uz leju līdz divām decimālajām zīmēm aiz komata.

Īsāk, ja jūsu programma izdara ne vairāk kā Q svēršanas katras izpildes reizes katrā testā, tad tā iegūs 100 punktus.

Piemērs

Pieņemsim, ka monētu secība no vieglākās līdz smagākajai ir **3 4 6 2 1 5**.

| Funkcijas izsaukums | Atgriež | Paskaidrojums |
|-----------------------------|---------|--|
| getMedian(4, 5, 6) | 6 | Monēta 6 ir ar vidējo svaru starp monētām 4, 5 un 6 . |
| getHeaviest(3, 1, 2) | 1 | Monēta 1 ir smagākā starp monētām 1, 2 un 3 . |
| getNextLightest(2, 3, 4, 5) | 3 | Monētas 2, 3 un 4 visas ir vieglākas par 5 -to monētu, tātad vieglākā no tām (3) tiek atgriezta. |
| getNextLightest(1, 6, 3, 4) | 6 | Monētas 1 un 6 abas ir smagākas par monētu 4 . No monētām 1 un 6 , monēta 6 ir vieglākā. |
| getHeaviest(3, 5, 6) | 5 | Monēta 5 ir smagākā starp monētām 3, 5 un 6 . |
| getMedian(1, 5, 6) | 1 | Monēta 1 ir ar vidējo svaru starp monētām 1, 5 un 6 . |
| getMedian(2, 4, 6) | 6 | Monēta 6 ir ar vidējo svaru starp monētām 2, 4 un 6 . |
| answer([3, 4, 6, 2, 1, 5]) | | Programma atrada pareizo atbildi šim testam. |

Piemēra vērtēšanas programma

Piemēra vērtēšanas programma ielasa ievaddatus šādā formā:

- 1.rindā: T — Testu skaits
- katrā no rindām no 2-tās līdz $T + 1$ -ajai: sešu dažādu skaitļu virkne no **1** līdz **6**: monētu secība no vieglākās līdz smagākajai.

Piemēram, ja ievaddatos ir divi testi, kuros monētu secība ir **1 2 3 4 5 6** un **3 4 6 2 1 5**, tad ievaddati ir šādi:

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

Piemēra testēšanas programma izdrukā masīvu, kas tika padots funkcijai `answer()`.