

Vaa'at

Aminalla on kuusi kolikkoa, jotka on numeroitu **1, 2, . . . , 6**. Hän tietää, että kaikilla kolikoilla on eri paino. Hän haluaisi järjestää kolikot niiden painon mukaan. Tätä varten hän on kehittänyt uudenlaisen vaa'an.

Perinteisessä vaa'assa on kaksi kuppia. Vaakaa käytetään niin, että kumpaankin kuppiin laitetaan kolikko, minkä jälkeen vaaka kertoo, kumpi kolikko on painavampi.

Aminan uusi vaaka on monimutkaisempi. Siinä on neljä kuppia, joiden nimet ovat **A, B, C** ja **D**. Vaa'assa on neljä erilaista mittaustapaa, joista jokainen vastaa eri kysymykseen kolikoita koskien. Kun Amina käyttää vaakaa, hänen täytyy laittaa tarkalleen yksi kolikko jokaiseen kupeista **A, B** and **C**. Lisäksi neljännessä mittaustavassa hänen täytyy myös laittaa tarkalleen yksi kolikko kuppiin **D**.

Neljässä mittaustavassa vaaka vastaa seuraaviin kysymyksiin:

1. Mikä kolikko kupeissa **A, B** ja **C** on painavin?
2. Mikä kolikko kupeissa **A, B** ja **C** on kevein?
3. Mikä kolikko kupeissa **A, B** ja **C** on mediaani? (Tämä on kolikko, joka ei ole painavin eikä kevein kolmesta kolikosta.)
4. Tarkastellaan kupeissa **A, B** ja **C** olevista kolikoista vain niitä, jotka ovat painavampia kuin kupin **D** kolikko. Jos tällaisia kolikoita on olemassa, mikä näistä kolikoista on kevein? Jos taas tällaisia kolikoita ei ole, mikä kolikoista kupeissa **A, B** ja **C** on kevein?

Tehtävä

Tehtäväsi on kirjoittaa ohjelma, joka järjestää Aminan kuusi kolikkoa niiden painon mukaan. Ohjelma voi käyttää Aminan vaakaa kolikon painojen vertailuun. Ohjelmallesi annetaan ratkaistavaksi joukko testitapauksia, joista jokainen vastaa tiettyä kolikkojoukkoa.

Ohjelmasi tulee toteuttaa funktiot `init` ja `orderCoins`. Jokaisen ohjelmasi suorituksen aikana arvostelija kutsuu ensin funktiota `init` yhden kerran. Tämä kertoo sinulle testitapausten määrän sekä mahdollistaa muuttujien alustamisen. Arvostelija kutsuu sitten funktiota `orderCoins()` kerran jokaiselle testitapaukselle.

- `init(T)`
 - `T`: Montako testitapausta ohjelmasi tulee ratkaista suorituksen aikana. `T` on kokonaisluku väliltä **1, . . . , 18**.
 - Funktiolla ei ole palautusarvoa.
- `orderCoins()`
 - Funktiota kutsutaan kerran kullekin testitapaukselle.

- Funktion tulee selvittää Aminan kolikoiden oikea järjestys käyttäen arvostelijan funktioita `getHeaviest()`, `getLightest()`, `getMedian()` ja/tai `getNextLightest()`.
- Kun funktio tietää oikean järjestyksen, sen tulee ilmoittaa se kutsumalla funktiota `answer()`.
- Funktion `answer()` kutsumisen jälkeen funktion `orderCoins()` tulee päättyä. Funktiolla ei ole palautusarvoa.

Voit käyttää seuraavia arvostelijan funktioita ohjelmassasi:

- `answer(W)` — ohjelmasi tulee käyttää tätä funktiota ilmoittamaan löytämänsä vastauksen.
 - `W`: 6-alkioinen taulukko, jossa on kolikoiden oikea järjestys. Alkioiden `W[0]`:sta `W[5]`:een tulee olla kolikoiden numerot (eli numerot **1, 2, ..., 6**) järjestyksessä keveimmästä raskaimpaan kolikkoon.
 - Ohjelmasi tulee kutsua tätä funktiota funktiosta `orderCoins()` kerran jokaiselle testitapaukselle.
 - Funktiolla ei ole palautusarvoa.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — nämä vastaavat mittaustapoja 1, 2 ja 3 Aminan vaa'assa.
 - `A, B, C`: Kolikot laitetaan kuppeihin ***A, B*** ja ***C***. `A, B` ja `C` ovat kolme eri kokonaislukua väliltä **1, 2, ..., 6**.
 - Kukin funktio palauttaa yhden numeroista `A, B` ja `C`: tulosta vastaavan kolikon numeron. Esimerkiksi `getHeaviest(A, B, C)` palauttaa annetuista kolikoista raskaamman numeron.
- `getNextLightest(A, B, C, D)` — tämä vastaa mittaustapaa 4 Aminan vaa'assa
 - `A, B, C, D`: Kolikot laitetaan kuppeihin ***A, B, C*** ja ***D***. `A, B, C` ja `D` ovat neljä eri kokonaislukua väliltä **1, 2, ..., 6**.
 - Funktio palauttaa yhden numeroista `A, B` ja `C`: mittaustavan 4 mukaisen kolikon numeron. Tämä tarkoittaa, että palautettu kolikko on kevein niiden kolikoiden joukossa, jotka ovat kupeissa ***A, B*** ja ***C*** sekä raskaampia kuin kupissa ***D*** oleva kolikko. Jos mikään niistä ei ole raskaampi kuin kupin ***D*** kolikko, palautettava kolikko on kevein kaikista kolmesta kolikosta kupeissa ***A, B*** ja ***C***.

Pisteytys

Tässä tehtävässä ei ole alitehtäviä. Sen sijaan pistemääräsi perustuu siihen, kuinka monta punnitusta (yhteismäärä kutsuja funktioihin `getLightest()`, `getHeaviest()`, `getMedian()` ja/tai `getNextLightest()`) ohjelmasi tekee.

Ohjelmasi suoritetaan useita kertoja, ja joka kerralla on useita testitapauksia. Olkoon r ohjelmasi suorituskertojen määrä. Tämä on kiinteä arvo testiaineistossa. Jos yhdenkin suorituskerran yhdessäkin testitapauksessa ohjelmasi ei järjestä kolikoita oikein, se saa 0 pistettä. Muuten suorituskerrat pisteytetään yksilöllisesti seuraavasti.

Olkoon Q pienin sellainen luku, että on mahdollista järjestää mikä tahansa kuuden kolikon joukko

käyttäen Q punnitusta Aminan vaa'alla. Jotta tehtävä olisi haastavampi, emme paljasta arvoa Q tässä.

Olkoon suurin määrä punnituksia kaikkien suorituskertojen kaikkien testitapausten joukossa $Q + y$, missä y on jokin kokonaisluku. Tarkastellaan nyt yksittäistä ohjelman suorituskertaa. Olkoon suurin punnitusten määrä kaikkien tämän suorituskerran T testitapausten joukossa $Q + x$, missä x on jokin ei-negatiivinen kokonaisluku. (Jos käytät alle Q punnitusta joka tapauksessa, niin $x = 0$.) Silloin tämän suorituskerran tulos on $\frac{100}{r((x+y)/5+1)}$ pyöristäen *alaspäin* kahden desimaalin tarkkuudelle.

Erityisesti jos ohjelmasi tekee korkeintaan Q punnitusta jokaisen suorituskerran joka testitapauksessa, saat tehtävästä 100 pistettä.

Esimerkki

Oletetaan, että kolikoiden järjestys on **3 4 6 2 1 5** keveimmästä raskaimpaan.

Funktiokutsu	Palauttaa	Selitys
getMedian(4, 5, 6)	6	Kolikko 6 on mediaani kolikkojen 4 , 5 ja 6 joukossa.
getHeaviest(3, 1, 2)	1	Kolikko 1 on raskain kolikoiden 1 , 2 ja 3 joukossa.
getNextLightest(2, 3, 4, 5)	3	Kolikot 2 , 3 ja 4 ovat kaikki keveämpiä kuin kolikko 5 , joten niistä kevein (3) palautetaan.
getNextLightest(1, 6, 3, 4)	6	Kolikot 1 ja 6 ovat raskaampia kuin kolikko 4 . Kolikoiden 1 ja 6 joukossa kolikko 6 on kevein.
getHeaviest(3, 5, 6)	5	Kolikko 5 on raskain kolikoiden 3 , 5 ja 6 joukossa.
getMedian(1, 5, 6)	1	Kolikko 1 on mediaani kolikoiden 1 , 5 ja 6 joukossa.
getMedian(2, 4, 6)	6	Kolikko 6 on mediaani kolikoiden 2 , 4 ja 6 joukossa.
answer([3, 4, 6, 2, 1, 5])		Ohjelma on löytänyt oikean vastauksen testitapaukseen.

Esimerkkiarvostelija

Esimerkkiarvostelija lukee syötettä seuraavassa muodossa:

- rivi **1**: T — testitapausten määrä
- jokainen riveistä **2**, \dots , $T + 1$: **6** eri numeroa väliltä **1**, **2**, \dots , **6**: kolikoiden järjestys keveimmästä raskaimpaan.

Esimerkiksi syöte, jossa on kaksi testitapausta, joissa kolikoiden järjestykset ovat **1 2 3 4 5 6** ja **3 4 6 2 1 5**, näyttää seuraavalta:

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

Esimerkkiarvostelija tulostaa taulukon, joka annettiin parametrina funktiolle `answer()`.