



Kaalud

Aminal on kuus münti, mis on nummerdatud **1** kuni **6**. Ta teab, et kõik mündid on erineva raskusega ja ta tahab need kõik raskuse järgi ära sorteerida. Selleks on ta leiutanud uut sorti kangkaalu.

Tavalisel kangkaalul on kaks kaalukaussi. Sellise kaalu kasutamisel pannakse kummalegi kaalukaussile üks münt ning kaal näitab, kumb münt on raskem.

Amina uus kaal on keerulisem. Sellel on neli kaalukaussi, mis on tähistatud **A**, **B**, **C** ja **D**. Kaalul on neli erinevat operatsiooni, millest igaüks vastab erinevale müntide kohta esitatud küsimusele. Kaalu kasutamiseks peab Amina panema täpselt ühe münti kaalukaussidele **A**, **B** ja **C**. Neljanda operatsiooni puhul peab panema ka täpselt ühe münti kaalukaussile **D**.

Need neli operatsiooni vastavad järgmistele neljale küsimusele:

1. Milline müntidest kaalukaussidel **A**, **B** ja **C** on raskeim?
2. Milline müntidest kaalukaussidel **A**, **B** ja **C** on kergeim?
3. Milline müntidest kaalukaussidel **A**, **B** ja **C** on keskmine? (Mitte raskeim ega kergeim nendest kolmest müntidest.)
4. Vaatleme müntidest kaalukaussidel **A**, **B** ja **C** ainult neid, mis on raskemad, kui münt kaalukaussil **D**. Kui selliseid münte leidub, siis milline neist on kergeim? Kui selliseid münte ei leidu, siis milline kõigist müntidest kaalukaussidel **A**, **B** ja **C** on kergeim?

Ülesanne

Kirjuta programm, mis sorteerib Amina kuus münti raskuse järgi. Programm saab kasutada Amina kaalu müntide raskuste võrdlemiseks. Programmile antakse ette mitu testjuhtumit, millest igaüks vastab uuele müntide komplektile.

Programm peab realiseerima funktsioonid `init` ja `orderCoins`. Programmi igal käivitamisel kutsub hindaja täpselt ühe korra välja funktsiooni `init`. See annab sulle võimaluse vajalike muutujate initsialiseerimiseks. Seejärel kutsub hindaja iga testjuhtumi jaoks ühe korra välja funktsiooni `orderCoins()`.

- `init(T)`
 - `T`: Testjuhtumite arv, mis tuleb lahendada. `T` on täisarv vahemikus **1, ..., 18**.
 - Funktsioon ei tagasta mingit väärtust.
- `orderCoins()`
 - Seda funktsiooni kutsutakse välja täpselt üks kord iga testjuhtumi jaoks.
 - Funktsioon peab leidma Amina müntide raskuse järjekorra, kasutades hindaja funktsioone

`getHeaviest()`, `getLightest()`, `getMedian()` ja `getNextLightest()`.

- Kui funktsioon teab õiget järjekorda, peab ta kutsuma välja hindaja funktsiooni `answer()`.
- Pärast funktsiooni `answer()` väljakutsumist peab funktsioon `orderCoins()` töö lõpetama, ilma mingit väärtust tagastamata.

Oma programmis saab kasutada järgmisi funktsioone:

- `answer(W)` — programm peaks kasutama seda funktsiooni saadud vastuse teatamiseks.
 - W : 6 elemendiga massiiv, milles on müntide õige järjekord. $W[0]$ kuni $W[5]$ peavad sisaldama müntide järjenumbreid (s.t arve **1** kuni **6**) kergetest raskeimani.
 - Programm peab kutsuma seda funktsioonist `orderCoins()`, üks kord igas testjuhtumis.
 - Funktsioon ei tagasta mingit väärtust.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — vastavad Amina kaalu operatsioonidele 1, 2 ja 3.
 - A, B, C : Mündid, mida panna vastavalt kaalukaussidele **A**, **B** ja **C**. A, B ja C peavad olema erinevate müntide järjenumbrid, igaüks lõigus **1** kuni **6**.
 - Iga funktsioon tagastab ühe arvudest A, B ja C : õige müntide järjenumbri. Näiteks `getHeaviest(A, B, C)` tagastab etteantud kolmest müntidest raskeima järjenumbri.
- `getNextLightest(A, B, C, D)` — vastab Amina kaalu operatsioonile 4.
 - A, B, C, D : Mündid, mis pannakse vastavalt kaalukaussidele **A**, **B**, **C** ja **D**. A, B, C ja D peavad olema erinevate müntide järjenumbrid, igaüks lõigus **1** kuni **6**.
 - Funktsioon tagastab ühe arvudest A, B ja C : operatsioonile 4 vastava müntide järjenumbri. See tähendab, et tagastatav number vastab kergetele müntidele **A**, **B** ja **C**, mis on raskemad, kui münt kaalukaussil **D** või kui ükski neist pole raskem kui münt kaalukaussil **D**, vastab tagastatav number lihtsalt kergetele müntidele kaalukaussidel **A**, **B** ja **C**.

Hindamine

Sellel ülesandel ei ole alamülesandeid. Selle asemel hinnatakse programmi selle järgi, mitu kaalumist (funktsioonide `getLightest()`, `getHeaviest()`, `getMedian()` ja `getNextLightest()` väljakutset) programm kokku teeb.

Programmi käivitatakse mitu korda, iga kord mitme testjuhtumiga. Olgu r programmi käivitamiste arv (fikseeritud). Kui programm eksib mõne järjestamisega mistahes testjuhtumis ja mistahes käivitamisel, saab ta 0 punkti. Vastasel korral hinnatakse programmi järgmiselt:

Olgu Q vähim teoreetiliselt võimalik arv kaalumisi, millega on võimalik Amina kaalu abil suvalist kuue müntide järjekorda õigesti sorteerida. Et ülesanne raskem oleks, siis me ei ütle praegu, mis Q väärtus on.

Olgu suurim kaalumiste arv kõigis käivitamistes ja kõigis testjuhtumites $Q + y$. Vaatleme nüüd üksikut käivitamist. Olgu suurim kaalumiste arv selle käivitamise T testjuhtumise seas $Q + x$. (Kui sa kasutad igas testjuhtumis vähem kui Q kaalumist, loetakse $x = 0$.) Siis on selle käivitamise skoor

$\frac{100}{r((x+y)/5+1)}$, ümardatud *alla*, kuni kaks kohta pärast koma.

Kui programm kasutab igas testjuhtumis igal käivitamisel ülimalt Q kaalumist, saab ta 100 punkti.

Näide

Olgu mündid järjekorras **3 4 6 2 1 5**, kergeimast raskeimani.

Funktsiooni väljakutse	Tagastab	Selgitus
getMedian(4, 5, 6)	6	Münt 6 on keskmine müntidest 4 , 5 ja 6 .
getHeaviest(3, 1, 2)	1	Münt 1 on raskeim müntidest 1 , 2 ja 3 .
getNextLightest(2, 3, 4, 5)	3	Mündid 2 , 3 ja 4 on kõik kergemad, kui münt 5 , seega tagastatakse neist kergeim (3).
getNextLightest(1, 6, 3, 4)	6	Mündid 1 ja 6 on raskemad, kui münt 4 . Müntidest 1 ja 6 on münt 6 kergem.
getHeaviest(3, 5, 6)	5	Münt 5 on müntidest 3 , 5 ja 6 raskeim.
getMedian(1, 5, 6)	1	Münt 1 on müntidest 1 , 5 ja 6 keskmine.
getMedian(2, 4, 6)	6	Münt 6 on müntidest 2 , 4 ja 6 keskmine.
answer([3, 4, 6, 2, 1, 5])		Programm leidis selle testjuhtumi õige vastuse.

Näidishindaja

Näidishindaja loeb sisendit järgmisel kujul:

- rida **1**: T — testjuhtumite arv
- read **2** kuni $T + 1$: järjend **6** erinevast arvust **1** kuni **6**: müntide järjestus kergeimast raskeimani.

Näiteks sisend, milles on kaks testjuhtumit müntide järjestustega **1 2 3 4 5 6** ja **3 4 6 2 1 5** kirjutatakse nii:

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

Näidishindaja kirjutab välja massiivi, mis anti parameetrina funktsioonile `answer()`.