

Datatähti 2015

A: Omenat

Aikaraja: 2 s

Maahan on pudonnut omenoita, ja Uolevi aikoo poimia niitä. Tiedät jokaisesta omenasta, kuinka painava se on.

Uolevi haluaa saada mahdollisimman monta omenaa, mutta hänellä on yksi lisäehto: omenoiden yhteispaino tulee olla 3:lla jaollinen.

Tehtäväsi on laskea, kuinka monta omenaa Uolevi voi poimia.

Syöte

Syötteen ensimmäisellä rivillä on kokonaisluku n : omenoiden määrä.

Seuraavalla rivillä on n kokonaislukua p_1, p_2, \dots, p_n : kunkin omenan paino.

Tuloste

Ohjelmasi tulee tulostaa yksi kokonaisluku: montako omenaa Uolevi pystyy poimimaan enintään.

Esimerkki

Syöte:

5

6 7 6 2 1

Tuloste:

4

Selitys: Yksi ratkaisu on, että Uolevi poimii kaikki omenat paitsi 7:n painoisen. Niiden yhteispaino on 15, joka on 3:lla jaollinen.

Osatehtävä 1 (17 pistettä)

- $1 \leq n \leq 20$
- $1 \leq p_k \leq 100$

Osatehtävä 2 (30 pistettä)

- $1 \leq n \leq 100$
- $1 \leq p_k \leq 100$

Osatehtävä 3 (53 pistettä)

- $1 \leq n \leq 10^5$
- $1 \leq p_k \leq 10^9$

Datatähti 2015

B: Urkupillit

Aikaraja: 2 s

Eräänä sunnuntaina Uolevi meni kirkkoon, ja urut tekivät häneen suuren vaikutuksen. Uolevi pani merkille, että urkupillejä oli rivissä n kappaletta ja niiden korkeudet olivat $1, 2, \dots, n$. Lisäksi Uolevi havaitsi, että uruissa oli yhteensä k pilliparia, joissa vasemmalla oleva pilli oli korkeampi kuin oikealla oleva pilli.

Kotiin palattuaan Uolevi halusi rakentaa pienoismallin kirkon uruista, mutta ei enää muistanut tarkasti niiden rakennetta. Niinpä hän päätti tyytyä pienoismalliin, joka täyttäisi yllä mainitut ehdot. Tehtäväsi on auttaa Uolevia tässä.

Syöte

Syötteen ainoalla rivillä on kaksi kokonaislukua n ja k .

Tuloste

Ohjelmasi tulee tulostaa n kokonaislukua, jotka kuvaavat urkupillien järjestyksen.

Jos ratkaisuja on useita, voit tulostaa minkä tahansa niistä. Voit olettaa, että ratkaisu on aina olemassa.

Esimerkki

Syöte:

5 3

Tuloste:

2 1 4 5 3

Selitys: Pillipareissa $(2, 1)$, $(4, 3)$ ja $(5, 3)$ vasemmalla oleva pilli on korkeampi kuin oikealla oleva pilli.

Rajat

$$\bullet 0 \leq k \leq \frac{n \cdot (n-1)}{2}$$

Osatehtävä 1 (12 pistettä)

- $1 \leq n \leq 5$

Osatehtävä 2 (33 pistettä)

- $1 \leq n \leq 100$

Osatehtävä 3 (55 pistettä)

- $1 \leq n \leq 10^5$

Datatähti 2015

C: Ruudukko

Aikaraja: 3 s

Uolevi on ruudukon vasemmassa yläkulmassa ja kulkee ruudukon oikeaan alakulmaan. Hän voi liikkua ruudukossa kerrallaan yhden ruudun alaspäin tai oikealle.

Jokaisessa ruudussa on kirjain, joista Uolevi muodostaa merkkijonon matkan aikana. Uolevi ottaa jokaisen reitin varrella olevan kirjaimen mukaan merkkijonoon. Mikä on aakkosjärjestyksessä pienin merkkijono, jonka Uolevi voi muodostaa?

Syöte

Syötteen ensimmäisellä rivillä on kokonaisluku n : ruudukon koko.

Tämän jälkeen syötteessä on n riviä, joista jokaisella on n merkkiä. Nämä rivit kuvaavat ruudukon kirjaimet. Jokainen kirjain on välillä A..Z.

Tuloste

Ohjelmasi tulee tulostaa yksi merkkijono: aakkosjärjestyksessä pienin merkkijono, jonka Uolevi pystyy muodostamaan.

Esimerkki

Syöte:

3

AAA

BAD

CCB

Tuloste:

AAACB

Osatehtävä 1 (17 pistettä)

- $1 \leq n \leq 5$

Osatehtävä 2 (42 pistettä)

- $1 \leq n \leq 100$

Osatehtävä 3 (41 pistettä)

- $1 \leq n \leq 500$

Datatähti 2015

D: Lukujono

Aikaraja: 4 s

Fibonaccin lukujono (F_n) on erittäin tunnettu lukujono, ja se määritellään seuraavasti:

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$, jos $n \geq 2$.

Paljon vähemmälle huomiolle on jäänyt Uolevinaccin lukujono (U_n), joka määritellään seuraavasti:

- $U_0 = 0$
- $U_1 = 1$
- $U_n = U_{\lfloor n/2 \rfloor} + U_{\lfloor n/3 \rfloor} + U_{\lfloor n/4 \rfloor} + U_{\lfloor n/5 \rfloor} + \dots + U_{\lfloor n/n \rfloor}$, jos $n \geq 2$.

Merkintä $\lfloor x \rfloor$ tarkoittaa lukua x pyöristettynä alaspäin kokonaisluvuksi.

Syöte

Syötteenä annetaan yksi kokonaisluku n .

Tuloste

Tulosteen tulee olla luku U_n .

Voit olettaa, että U_n on enintään 10^{18} .

Esimerkki

Syöte:

10

Tuloste:

13

Selitys: $U_{10} = U_5 + U_3 + U_2 + U_2 + U_1 + U_1 + U_1 + U_1 + U_1$.

Osatehtävä 1 (16 pistettä)

- $1 \leq n \leq 10^3$

Osatehtävä 2 (34 pistettä)

- $1 \leq n \leq 10^5$

Osatehtävä 3 (50 pistettä)

- $1 \leq n \leq 10^8$

Datatähti 2015

E: Pörssihai

Aikaraja: 5 s

Uolevi on sijoittanut rahaa erään firman osakkeisiin ja seuraa kurssin kehitystä päivittäin. Maija epäilee kuitenkin, onko touhussa järkeä.

Eräänä päivänä Uolevi päätti todistaa Maijalle, että sijoitus on ollut kannattava, kunhan tarkastellaan sopivaa ajanjaksoa. Sitä varten Uolevi tarvitsee kuitenkin apuasi.

Kun annettuna on osakkeen hinta n päivän aikana, tehtäväsi on etsiä joka päivälle sellainen kyseiseen päivään päättyvä ajanjakso, että hinnan keskiarvo on siinä mahdollisimman suuri.

Syöte

Syötteen ensimmäisellä rivillä on kokonaisluku n : päivien määrä.

Tämän jälkeen syötteessä on n kokonaislukua a_1, a_2, \dots, a_n : osakkeen arvo kunakin päivänä.

Tuloste

Ohjelmasi tulee tulostaa n kokonaislukua: kuhunkin päivään päättyvän ajanjakson pituus, jossa osakkeen hinnan keskiarvo on korkein.

Jos mahdollisuuksia on monia, tulosta mahdollisimman pitkä ajanjakso.

Esimerkki

Syöte:

7

1 6 4 6 2 5 5

Tuloste:

1 1 2 1 4 1 2

Selitys: Tarkastellaan esimerkiksi päivää, jolloin osakkeen arvo oli 2. Tähän päivään päättyvät keskiarvot ovat $\frac{1+6+4+6+2}{5} = 3.8$, $\frac{6+4+6+2}{4} = 4.5$, $\frac{4+6+2}{3} = 4$, $\frac{6+2}{2} = 4$ ja $\frac{2}{1} = 2$. Näistä korkein keskiarvo saadaan valitsemalla ajanjakso 4 päivää.

Osatehtävä 1 (12 pistettä)

- $1 \leq n \leq 200$
- $1 \leq a_k \leq 10^6$

Osatehtävä 2 (17 pistettä)

- $1 \leq n \leq 5000$
- $1 \leq a_k \leq 10^6$

Osatehtävä 3 (71 pistettä)

- $1 \leq n \leq 10^6$
- $1 \leq a_k \leq 10^6$

Datatähti 2015

F: Kartta

Aikaraja: 5 s

Vuosisatojen aikana merenpinta Uolevin asuinseudulla on kohonnut. Kun perinnepäivät koittivat, Uolevi pyydettiin pitämään esitelmä aiheesta.

Uolevilla on käytössä korkeuskartta seudusta, ja hän haluaa pystyä näyttämään jokaiselle korkeudelle x kuvan, jossa sinisellä on merkitty alueet, joiden korkeus on enintään $x - 1$, ja vihreällä alueet, joiden korkeus on x tai enemmän.

Ongelmana on kuitenkin, että Uolevilla on käytössään vain dokumenttikamera, 40 kalvoa sekä vihreää ja sinistä läpinäkymätöntä maalia. Tehtäväsi on päättää, miten Uolevin tulisi maalata kalvot, jotta hän voi mille tahansa korkeudelle x muodostaa kalvoista oikeanlaisen kuvan.

Uolevi muodostaa kalvoista kuvan valitsemalla joukon kalvoja ja pinoamalla ne johonkin järjestykseen. Tämän jälkeen jokaisessa pisteessä näkyy väri, joka on mahdollisimman korkealla pinossa olevassa kalvossa kyseisessä pisteessä. Jos pisteen kohdalla missään kalvossa ei ole väriä, kuva on virheellinen.

Uolevi pinoaa kalvot tarkalleen päällekkäin eikä hän muuta niiden asentoa.

Syöte

Ensimmäisellä rivillä on kolme kokonaislukua: korkeuskartan korkeus n ja leveys m sekä suurin mahdollinen korkeus h .

Seuraavat n riviä sisältävät korkeuskartan: kukin rivi sisältää sen rivin korkeudet (m kokonaislukua väliltä $1 \dots h$).

Tuloste

Ohjelmasi tulee tulostaa ensin, miltä kukin 40 kalvosta näyttää. Jokaista kalvoa kohden tulosteessa tulee olla n riviä, joilla on m merkkiä. Jokainen rivi koostuu merkeistä S (sininen), V (vihreä) ja L (läpinäkyvä). Kalvot on numeroitu kokonaisluvuin $1, 2, \dots, 40$.

Sitten ohjelman tulee tulostaa h riviä: jokaiselle korkeudelle $x = 1, 2, \dots, h$ jokin tapa esittää kuva kalvojen avulla. Rivin alussa on käytettävien kalvojen määrä, minkä jälkeen rivillä on kalvojen numerot järjestyksessä pinon pohjalta ylös. Samaa kalvoa saa käyttää vain kerran.

Esimerkki

Syöte:

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

Tuloste:

```
VVVVV
VVVVV
VVVVV
SSLSL
LSSSL
SSLSL
LLSLS
LLLLL
LLLLL
LLLLL
LLLLL
SLLLS
LLLLS
LLLLL
LLLLL
LLSLL
...
1 1
2 1 2
3 1 2 3
4 1 2 3 4
```

Merkinnän ... kohdalle kuuluu 35 mitä tahansa kalvon kuvausta (niitä ei tarvita kuvien muodostamisessa tässä syötteessä).

Rajat

- $1 \leq n, m \leq 100$

Osatehtävä 1 (12 pistettä)

$$1 \leq h \leq 40$$

Osatehtävä 2 (27 pistettä)

$$1 \leq h \leq 200$$

Osatehtävä 3 (15 pistettä)

$$1 \leq h \leq 1000$$

Osatehtävä 4 (46 pistettä)

$$1 \leq h \leq 10^4$$

Datatähti 2015

Tekniset ohjeet

Rajat

Jokaisen tehtävän aikaraja on ilmoitettu tehtävän alussa.

Jokaisen tehtävän muistiraja on 128 Mt. Pinomuistia saa käyttää enintään 64 Mt.

Koodin koko saa olla enintään 128 kt.

Ohjelmointikielet

Sallitut ohjelmointikielet ovat C++ ja Java. Huomaa, että voit joutua käyttämään alla olevia optimointeja, jotta syötteen lukeminen ja tulostus on riittävän nopeaa.

C++

C++ käännetään g++ 4.8.2-kääntäjällä. Kääntäjäliput ovat `-std=c++0x -O2 -Wall`.

C++:n `cin` ja `cout` voivat olla oletuksena liian hitaita, mutta niitä nopeuttaa huomattavasti, jos poistaa niiden synkronoinnin C:n `stdio`-kirjaston kanssa. Tämä tehdään lisäämällä rivi

```
cin.sync_with_stdio(false);
```

main-funktion alkuun:

```
#include <iostream>

using namespace std;

int main() {
    cin.sync_with_stdio(false);

    // ratkaise tehtävä tähän
}
```

Java

Käytettävä Java-jakelu on OpenJDK versio 1.8.0_40 (eli siis Java 8).

Nimeä kooditiedostosi Javan käytäntöjen mukaisesti, eli jos main-luokan nimi on Main niin uploadattavan tiedoston nimen tulee olla Main.java. Koodi ei saa olla missään pakkauksessa, tai muuten käänntö epäonnistuu.

Syötteeseen ja tulostukseen Javan `Scanner`-luokka ja komento `System.out.println` saattavat olla liian hitaita. Ajoympäristössä on käytössäsi nopeamman syötteen ja tulostuksen mahdollistava IO-luokka, jota ei tarvitse erikseen importata (`IO.class` on samassa hakemistossa ohjelman kanssa). IO-luokkaa käytetään seuraavasti:

```
public class Ratkaisu {
public static void main(String[] args) {
    IO io = new IO();

    String a = io.next(); // Lukee seuraavan välein erotellun merkkijonon.
    int b = io.nextInt(); // Lukee seuraavan välein erotellun int-kokonaisluvun.
    long c = io.nextLong(); // Lukee seuraavan välein erotellun long-kokonaisluvun.
    double d = io.nextDouble(); // Lukee seuraavan välein erotellun double-liukuluvun.

    // Toimii kuten System.out.println.
    io.println("Annoit syötteenä " + a + " " + b + " " + c + " " + d);

    io.close(); // TÄYTYY KUTSUA LOPUKSI, muuten tuloste voi jäädä kirjoittamatta
}
}
```