
Datatähti 2010 -alkukilpailu

Ohjelmointitehtävä 1/3:

Mustikat

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Tarkastellaan seuraavaa kahden pelaajan peliä: Aluksi pöydällä on kasa mustikoita. Tämän jälkeen pelaajat syövät vuorotellen yhden, kolme tai neljä mustikkaa pöydältä. Viimeisen mustikan syöjä voittaa pelin.

Tehtävänä on selvittää, kummalla pelaajalla on voittostrategia eli kumpi pelaaja voittaa, jos molemmat pelaavat parhaalla mahdollisella tavalla. Jos pelaajalla on voittostrategia, hän voittaa pelin varmasti vastustajan menettelystä riippumatta. Tässä pelissä jommallakummalla pelaajalla on aina voittostrategia.

Esimerkki 1: Pöydällä on aluksi kuusi mustikkaa. Nyt aloittaja voittaa, koska hän voi syödä neljä mustikkaa. Tämän jälkeen toisen pelaajan on pakko syödä yksi mustikka, minkä jälkeen aloittaja syö viimeisen mustikan ja voittaa pelin.

Esimerkki 2: Pöydällä on aluksi seitsemän mustikkaa. Nyt aloittaja häviää, koska jos hän syö yhden mustikan, pöydälle jää kuusi mustikkaa ja ensimmäisen esimerkin mukaisesti toinen pelaaja voittaa. Jos aloittaja syö kolme mustikkaa, pöydälle jää neljä mustikkaa ja toinen pelaaja voi syödä ne kaikki. Jos taas aloittaja syö neljä mustikkaa, pöydälle jää kolme mustikkaa ja toinen pelaaja voi syödä ne kaikki.

Syöte

Tiedosto **mustikat.in** sisältää yhden rivin, jossa lukee pöydällä olevien mustikoiden lukumäärä pelin alussa. Mustikoita on vähintään yksi ja korkeintaan miljardi.

Tuloste

Ohjelman täytyy kirjoittaa tiedostoon **mustikat.out**, kummalla pelaajalla on voittostrategia. Jos aloittaja voittaa, ohjelman täytyy kirjoittaa tiedostoon kirjain A. Jos toinen pelaaja voittaa, ohjelman täytyy kirjoittaa tiedostoon kirjain T.

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **mustikat.c**, **mustikat.cpp** tai **mustikat.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 32 megatavua keskusmuistia. Kunkin testin kohdalla oikea tuloste antaa 5 pistettä ja väärä 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1

mustikat.in:

6

mustikat.out:

A

Esimerkki 2

mustikat.in:

7

mustikat.out:

T

Datatähti 2010 -alkukilpailu

Ohjelmointitehtävä 2/3:

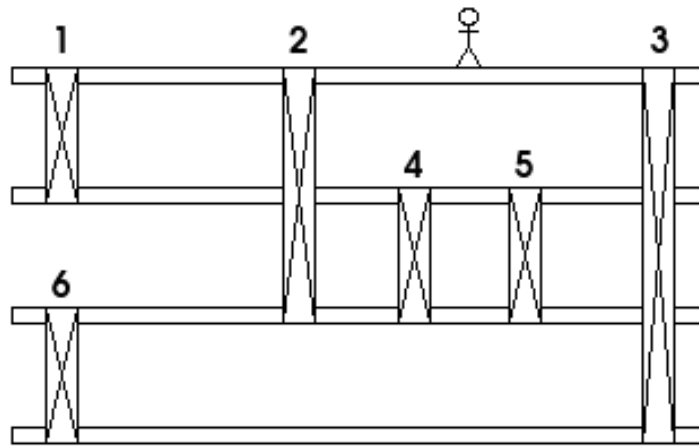
Liukumäkitalo

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Huvipuistossa on liukumäkitalo, joka muodostuu vaakasuorista kerroksista ja niitä yhdistävistä liukumäistä. Uolevi on liukumäkitalon ylimmässä kerroksessa. Miten monta tapaa hänellä on laskea alimpaan kerrokseen?

Jokainen liukumäki alkaa tietyistä kerroksesta ja päättyy tiettyyn kerrokseen. Liukumäkeä saa vain laskea alaspäin, sitä ei saa nousta ylöspäin. Liukumäki täytyy laskea alusta loppuun. Kerroksissa saa kulkea vapaasti laskujen välillä.

Seuraavassa esimerkissä kerroksia on neljä ja liukumäkiä on kuusi:



Nyt Uolevillä on neljä tapaa laskea alimpaan kerrokseen:

- liukumäki 1, liukumäki 4, liukumäki 6
- liukumäki 1, liukumäki 5, liukumäki 6
- liukumäki 2, liukumäki 6
- liukumäki 3

Voit olettaa, että kerroksia on korkeintaan tuhat ja liukumäkiä on korkeintaan miljoona. Lisäksi voit olettaa, että tapoja laskea alas on ainakin yksi ja korkeintaan 10^{18} .

Syöte

Ohjelmasi lukee syötteenä yhden ASCII-tekstitiedoston, jonka nimi on **liukumaki.in**.

Tiedoston **liukumaki.in** ensimmäisellä rivillä on kerrosten ja liukumäkien määrä. Sitten jokaista liukumäkeä vastaa yksi rivi, jossa lukee, mistä kerroksesta mihin kerrokseen liukumäki johtaa. Kerrokset on numeroitu järjestyksessä kokonaislukuvin: ylin kerros on 1, toiseksi ylin kerros on 2 jne.

Tuloste

Ohjelmasi kirjoittaa vastauksen uuteen ASCII-tiedostoon nimeltä **liukumaki.out**.

Ohjelman täytyy kirjoittaa tiedostoon **liukumaki.out**, kuinka monella tavalla Uolevi voi laskea liukumäkitalon ylimmästä kerroksesta alimpaan kerrokseen.

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **liukumaki.c**, **liukumaki.cpp** tai **liukumaki.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 0,1 sekuntia aikaa ja 32 megatavua keskusmuistia. Kunkin testin kohdalla täysin oikea tuloste antaa 5 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1

liukumaki.in:

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

liukumaki.out:

```
4
```

Esimerkki 2

liukumaki.in:

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

liukumaki.out:

```
1
```

Datatähti 2010 -alkukilpailu

Ohjelmointitehtävä 3/3:

Eristys

HUOM: Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 7)

Aapeli Amatööri teki remonttia, jonka yksi osa oli suorakulmion muotoisen seinän eristeen uusiminen. Aapelilla oli N kappaletta suorakulmion muotoisia ja mahdollisesti keskenään erikokoisia eristelevyjä, jotka hän sijoitti enemmän tai vähemmän satunnaisesti kohtiin seinälle. Sekä seinän että kunkin eristelevyn korkeus ja leveys voidaan ilmaista kokonaislukuina. Voimme siis kuvata sekä seinän että kunkin eristepalan yksikköruuduista muodostuvana suorakulmion muotoisena ruudukkona.

Seinä on A mittayksikköä leveä ja B mittayksikköä korkea eli $A \times B$ kokoinen yksikköneliöistä muodostuva ruudukko, jossa on A saraketta ja B riviä.

Kukin eristelevy on sijoitettu seinälle siten, että levyn yksikköruudut ja seinän yksikköruudut ovat kohdakkain. Kunkin eristelevyn sijoituskohta ilmaistaan antamalla sen seinän yksikköruudun koordinaatit, jonka kohdalla kyseisen eristelevyn vasemman yläkulman yksikköruutu sijaitsee. Koordinaatisto on sellainen, että seinän yksikkösarakkeet on numeroitu vasemmalta oikealle luvuin $1 \dots A$ ja yksikkörivit on numeroitu ylhäältä alas luvuin $1 \dots B$.

Seinälle sijoitetut levyt voivat asettua mahdollisesti kokonaan tai osittain toistensa kanssa päällekkäin. Seinän sarakkeessa x ja rivillä y eli ruudussa (x, y) sijaitsevan yksikköruudun eristyksen paksuus on sellaisten eristelevyjen lukumäärä, jotka peittävät yksikköruudun (x, y) .

Sijoiteltuaan kaikki N eristelevyä seinälle, Aapeli alkoi pohtia eristyksen laatua. Hän päätti laskea, kuinka monta seinän yksikköruutua omaa eristyksen, jonka paksuus on vähintään P . Auta häntä toteuttamalla ohjelma, joka saa syötteenään parametrit A , B , N ja P sekä N :n eristelevyn kuvauksen (mitat sekä sijoituskohdan), ja joka laskee, kuinka monta seinän yksikköruutua omaa vähintään P :n paksuisen eristyksen.

Syöte

Ohjelmasi lukee syötteenä yhden ASCII-tekstitiedoston, jonka nimi on **eristys.in**.

Syötetiedoston **eristys.in** ensimmäisellä rivillä on neljä toisistaan välilyönnein eroteltua positiivista kokonaislukua: luvut ovat järjestyksessä A , B , N ja P .

Tätä seuraa N riviä eli rivit $2 \dots 1 + N$. Rivi $1 + i$ kuvaa eristelevyn numero i neljällä toisistaan välilyönneillä erotellulla positiivisella kokonaisluvulla, jotka ovat järjestyksessä A_i , B_i , x_i ja y_i . Näistä A_i ja B_i ilmaisevat kyseisen eristelevyn leveyden ja korkeuden, ja x_i ja y_i ilmaisevat, että eristelevyn vasen yläkulma sijaitsee seinän ruudussa (x_i, y_i) . Kunkin eristelevyn sijoituskohta (x_i, y_i) on sellainen, että $x_i + A_i - 1 \leq A$ ja $y_i + B_i - 1 \leq B$ eli levy mahtuu kokonaisuudessaan seinän alueelle.

Edellä mainittujen seikkojen lisäksi syötteen luvuille pätevät seuraavat rajat:

$$1 \leq A, B \leq 20000$$

$$1 \leq N, P \leq 4000$$

$$1 \leq A_i \leq A$$

$$1 \leq B_i \leq B$$

Tuloste

Ohjelmasi kirjoittaa kelvollisten lukujen määrän uuteen ASCII-tiedostoon nimeltä **eristys.out**.

Tulostiedostoon **eristys.out** tulee kirjoittaa yksi rivinvaihtoon päättyvä rivi, joka sisältää yhden kokonaisluvun: niiden seinän yksikköruutujen lukumäärä, joiden eristyksen paksuus on vähintään P .

Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla ohjelmointikielestä riippuen **eristys.c**, **eristys.cpp** tai **eristys.pas**.

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 32 megatavua keskusmuistia. Kunkin testin kohdalla oikea tuloste antaa 5 pistettä ja väärä 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimi on siis 100 pistettä.

Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Esimerkki 1
<u>eristys.in:</u>
4 4 4 1
4 1 1 3
1 1 2 3
4 3 1 1
1 3 1 2
<u>eristys.out:</u>
13

Esimerkki 2
<u>eristys.in:</u>
10 10 5 3
7 6 3 3
5 1 2 8
4 8 4 1
4 10 7 1
7 8 1 1
<u>eristys.out:</u>
27

Esimerkkiä 2 havainnollistava kuva

Kuva esittää esimerkkisyötteen 2 seinän tilanteen siten, että kuhunkin ruutuun on merkitty sen eristyksen paksuus. Ne ruudut, joiden eristyksen paksuus on vähintään P , on korostettu.

1	1	1	2	2	2	3	1	1	1
1	1	1	2	2	2	3	1	1	1
1	1	2	3	3	3	4	2	2	1
1	1	2	3	3	3	4	2	2	1
1	1	2	3	3	3	4	2	2	1
1	1	2	3	3	3	4	2	2	1
1	2	3	4	4	4	4	2	2	1
0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	1	1	1

Sääntöjä ja ohjeita

Mitä palautetaan?

Kunkin ohjelmointitehtävien vastaus koostuu tehtävän yhteydessä mainitulla tavalla nimitystä yhdestä lähdekooditiedostosta. Siis jokaista tehtävää kohden oma tiedosto. Älä lähetä esimerkiksi käännettä ohjelmatiedostoa.

Sallitut ohjelmointikielet

Kunkin ohjelmointitehtävän vastauksessa sallitut ohjelmointikielet ovat C, C++ ja Pascal.

Lähdekooditiedostot

Lähdekooditiedoston nimen tiedostopäätte määräytyy seuraavasti:

kieli	tiedostopäätte
C	c
C++	cpp
Pascal	pas

Kunkin lähdekooditiedoston alussa pitää olla kommenttiosio, joka sisältää seuraavat tiedot:

1. Otsikkoteksti "Datatähti 2010 -alkukilpailu"
2. Tehtävän numero
3. Nimesi
4. Syntymäaikasi
5. Koulusi/lukiosi nimi ja paikkakunta

Ohjelmakoodin kirjoitusasu on muuten vapaa (esim. koodin selkeyttä ei arvostella).

Ohjelman toiminta

Ohjelmasi on noudatettava alla lueteltuja vaatimuksia, tai muuten seurauksena saattaa olla pistemenetyksiä.

- Ohjelma saa lukea ja kirjoittaa vain tehtävässä nimetyjä tiedostoja. Erityisesti näppäimistöä lukeminen tai näytölle tulostaminen on kiellettyä.
- Tulostiedoston täytyy aina päättyä rivinvaihtoon.
- Ohjelmointikielissä C ja C++ pääohjelman main suorituksen pitää päättyä lauseeseen `return 0;` (Pascal-kääntäjä huolehtii tästä puolestasi).
- Ohjelmointikielessä pitää rajoittua vain sen standardoituihin peruspiirteisiin, ja välttää laite- tai kääntäjäkohtaisia erikoisuuksia.
 - Ohjelmointikielissä C ja C++ on käytössä standardoitu matematiikkakirjasto (eli käännettäessä annetaan valitsin `-lm`).
 - Yleisesti ottaen saa käyttää ohjelmointikielten standardikirjastoja (esim. C++-kielen Standard Template Library (STL), C-kielen peruskirjasto (C standard library)).

Koodin ajoympäristö

Testitietokoneessa on Linux käyttöjärjestelmä sekä seuraavat kääntäjät:

Kieli	Kääntäjä	Versio	Kielistandardi (valitsimet)	Optimointitaso
C	gcc	≥ 4.0	ANSI/ISO C89/90 (-ansi)	-O2 -static
C++	g++	≥ 4.0	ISO C++ (-ansi)	-O2 -static
Pascal	Free Pascal	≥ 2.0	Borland Turbo Pascal 7.0 (-Mtp -Sg)	-O2 -XS

Resurssirajat

Tehtävien kohdalla todettiin, että ohjelma saa käyttää kunkin testiajon aikana vain tietyn määrän aikaa ja muistia. Voit arvioida testikoneen nopeutta sen pohjalta, että siinä on AMD Athlon64 3200+ CPU.

Ohjelmointitehtävien vastausten palautus

Jos samalta henkilöltä tulee kilpailuajana useita eri ratkaisuja, niin viimeinen niistä arvostellaan.

Jos kilpailija lähettää vastauksensa myöhästyneenä, voidaan vastaus jättää huomiotta.

Ratkaisujen palautus tapahtuu sähköpostitse osoitteeseen heikki.hyyro@cs.uta.fi. Viestin rakenne on seuraava:

- Sähköpostin otsikkokenttään (subject) teksti “Datatähti 2010 -alkukilpailun vastaukset”.
- Viestin rungoksi omat yhteystietosi:
 1. Nimesi
 2. Syntymäaikasi
 3. Koulusi/lukiosi nimi ja paikkakunta
 4. Postiosoitteesi
 5. Puhelinnumerosi
 6. Sähköpostiosoitteesi
- Lähdekooditiedostot laitetaan sähköpostin mukaan erillisessä zip-pakkauksessa. Zip-pakkaus tulee nimetä muodossa **etunimi_sukunimi.zip**.

– Tämän toiminnon nimi on yleensä “Liitä tiedosto”, “Attach File” tai vastaava.

Sähköpostivastauksen katsotaan lähteneen ajoissa, jos sen aikaleima on kilpailuajan sisällä. Tämä aikaleima katsotaan sähköpostin välittäneestä palvelinkoneesta.

Kysymyksiä (ja vastauksia)

Voit esittää sähköpostitse täsmennyskysymyksiä ohjelmointitehtäviin tai niiden palautukseen liittyen. Käytä edellä annettua vastausten palautusosoitetta. Kysymykset ja vastaukset laitetaan kilpailun aikana näkyviin kisasivustolle ilman kysyjien nimiä. Sivuja kannattaa siis seurata!