

0	5	7
4	6	2
8	1	3

Kuva 1: Eräs kokoa $N = 3$ oleva taikaneliö.

Datatähti-alkukilpailu 2005

1 Taikaneliö

Kokoa N oleva taikaneliö on N rivin ja N sarakkeen ruudukko, jonka ruutuihin on sijoitettu luvut $0, 1, 2, 3, \dots, N^2 - 1$ siten, että jokaisessa ruudussa on täsmälleen yksi luku, ja jokaisen vaaka- tai pystyrivin lukujen summa on aina sama. Esimerkiksi jos valitset kuvassa 1 minkä tahansa vaaka- tai pystyrivin, ja lasket sillä olevat luvut yhteen, niin saat aina saman tuloksen 12.

Taikaneliön voi tehdä kaikille muille N kuin 2 tai 6. Miten tekisit taikaneliön parittomille N ?

1.1 Tehtävä

Tehtävänäsi on kirjoittaa sellainen tietokoneohjelma, joka saa syötteekseen parittoman numeron N , ja tulostaa kokoa N olevan taikaneliön.

1.2 Syöttö

Ohjelmasi lukee syötteensä tekstitiedostosta nimeltä `taika.in`. Tiedoston sisältö on yksi kokonaisluku: kysytty koko N . Se on pariton ja väliltä $1 \dots 999$.

1.3 Tulostus

Ohjelmasi tulostaa tekstitiedostoon nimeltä `taika.out`. Tiedoston pitää sisältää yhteensä $N^2 + 1$ kokonaislukua seuraavasti:

- ensin kysytty taikaneliön koko N
- sitten taikaneliön ensimmäisen rivin luvut järjestyksessä vasemmalta oikealle
- sitten toisen rivin luvut järjestyksessä vasemmalta oikealle
- ja niin edelleen, kunnes koko taikaneliön kaikki luvut on lueteltu tässä järjestyksessä.

Kahden luvun välissä pitää olla joko tyhjämerkki tai rivinvaihtomerkki. Tulostiedoston muoto voi siis olla erilainen kuin taikaneliön, kunhan lukujen järjestys on molemmissa sama.

Esimerkiksi kuvan 1 taikaneliö voidaan tulostaa vaikkapa seuraavana tiedostona:

```
9 0 5 7 4
6 2 8 1 3
```

1.4 Pisteytys

Ohjelmaasi testataan 20 syötteellä. Näistä 10 syötteessä N on alle 100.

Ohjelmasi saa yhdelle syötteelle 64 megatavua työmuistia ja 1 sekunnin keskusyksikköaikaa. Ohjelmasi saa yhdestä syöttestä

5 pistettä jos sen suoritus päättyy normaalisti ylittämättä muisti- tai aikarajoja, ja tulostaa jonkin tälle syötteelle oikean vastauksen

0 pistettä muuten.

Työmuistiraja sisältää kaiken muun ohjelmasi käyttämän muistin paitsi sen suorituskelpoista ohjelmakoodia.

Keskusyksikköaikaraja sisältää sen ajan, jonka tietokoneen keskusyksikkö käyttää suorittaessaan sinun omaa ohjelmakoodiasi. Se ei sisällä sitä aikaa, jonka se käyttää suorittaessaan käyttöjärjestelmäpalveluita. Erityisesti se ei sisällä sitä aikaa, joka kuluu syötteen lukemiseen tiedostosta `taika.in` työmuistiin tai valmiin vastauksen tulostamiseen tiedostoon `taika.out`. Toisaalta se sisältää erilaisissa kirjastokutsuissa käytetyn ajan, esimerkiksi luetun syötteen muuntamisen tekstimuodosta vastaavaksi numeroksi.

Ohjelmat testataan sellaisella tietokoneella, jossa keskusyksikkönä on sellainen Intel Pentium P4 -mikroprosessori, jonka kellotaajuus on 3 GHz. Sen perusteella voit yrittää arvioida, kuinka pitkä vastaava keskusyksikköaikaraja on omalla koneellasi. Testikoneen käyttöjärjestelmänä on Linux.

1.5 Lähdekooditiedosto

Kirjoita ratkaisusi yhteen ASCII-lähdekooditiedostoon, jonka nimi riippuu käyttämästäsi ohjelmointikielestä seuraavasti:

KIELI	NIMI
C	<code>taika.c</code>
C++	<code>taika.cpp</code>
Pascal	<code>taika.pas</code>

Laita tiedoston alkuun kommentti, jossa on seuraavat asiat:

- Teksti "Datatähti-alkukilpailu 2005".
- Oma nimesi ja syntymäaikasi.
- Tämän tehtävän nimi "taika".
- Ilmoita, kääntyykö ohjelmasi kääntäjän oletusvalitsimilla. Jos ei, niin ilmoita lisäksi, mitkä valitsimet on lisättävä kääntäjän komentoriville. Ilmoita vain sellaiset valitsimet, joita ilman ohjelmasi ei käänny!

Tällainen välttämätön lisävalitsin voisi olla esimerkiksi matematiikkakirjastoa käytävissä C-kielisissä ohjelmissa tarvittava "-lm".

Muuten tiedoston kirjoitusasu on vapaa. Älä kuitenkaan käytä yli 80 merkkiä pitkiä rivejä!

Ohjelmointikielillä C ja C++ kirjoitetut lähdekooditiedostot käännetään kääntäjällä GCC <<http://gcc.gnu.org/>>. Ohjelmointikielellä Pascal kirjoitetut tiedostot käännetään puolestaan kääntäjällä Free Pascal <<http://www.freepascal.org/>>, jonka pitäisi olla Turbo Pascal

-yhteensopiva. Käytä kuitenkin vain valitsemasi ohjelmointikielen peruspiirteitä yhteensopivuuden takaamiseksi.

Käännökset tehdään kääntäjien oletusarvoisilla valitsimilla, ellei ole ilmoittanut muuta tiedoston alkukommentissa.

Huomioi vielä seuraavat tekniset vaatimukset:

- Ohjelmasi saa lukea syötettä vain tiedostosta nimeltä `taika.in` ja tulostaa vain tiedostoon nimeltä `taika.out`!

Erityisesti se *ei* saa lukea mitään näppäimistöltä tai tulostaa mitään näytölle.

- Kun ohjelmasi suoritus päättyy, niin sen täytyy palauttaa käyttöjärjestelmälle poistumisarvonaan 0. Muuten käyttöjärjestelmä katsoo, ettei ohjelmasi päättynytkään normaalisti vaan johonkin virheeseen.

Ohjelmointikielissä C ja C++ tämä tarkoittaa, että pääohjelman `main` (ja koko ohjelman) suorituksen pitää päättyä komentoon `return 0;` (eikä vain ”`valua`” viimeiseen sulkevaan aaltosulkuun `’}`’ saakka).

Ohjelmointikielessä Pascal kääntäjä huolehtii tästä automaattisesti.

Näiden vaatimusten rikkomisesta saattaa seurata pistemenetyksiä testiajojen epäonnistuessa!

Datatähti-alkukilpailu 2005

2 Pac-Man

”Jos videopelit oikeasti vaikuttaisivat nuoriimme, niin he harhailisivat pimeissä sokkeloissa monotonisen musiikin tahtiin popsimassa erivärisiä pillereitä.”

Eräs videopelien alkuaikojen suurimmista klassikoista oli *Pac-Man*, jossa pelaaja pakeni sokkelossa haamuja, jotka yrittivät syödä hänet. Sokkelossa oli siellä täällä voimapillereitä, ja sellaisen syötyään pelaajasta tuli takaa-ajaja, joka pystyikin syömään kohtaamiaan haamuja. Mutta vain hetkeksi: kohta voimapillerin vaikutus lakkasi, ja taas piti pelaajan juosta haamuja pakoon.

Kuinka kauan yhden voimapillerin vaikutuksen pitäisi kestää, jotta pelaaja olisi turvassa haamuilta?

2.1 Tehtävä

Tee ohjelma joka ratkaisee seuraavan ongelman: Syötteenä annetaan Pac-Man-pelin sokkelo. Pelaaja on aluksi laudan yhdessä nurkassa, jossa hän on juuri syönyt voimapillerin. Tulosteena halutaan pienin sellainen luku p , että jos yhden voimapillerin vaikutus kestää tasan p askelta, niin pelaaja voi päästä laudan vastakkaiseen nurkkaan siten, että hän ehtii aina seuraavan voimapillerin luokse ennen kuin edellisen vaikutus lakkaa.

Voimapillereitä ei voi syödä varastoon: Edellisen pillerin jäljellä oleva vaikutus katoaa, kun seuraava syödään.

2.2 Syöttö

Ohjelmasi lukee syötteesä tekstitiedostosta nimeltä `pacman.in`.

Tiedoston ensimmäinen rivi koostuu yhdestä kokonaisluvusta N , joka on väliltä $2 \dots 50$.

Sen jälkeen tiedosto sisältää N riviä. Jokaisella rivillä on N merkkiä seuraavasti:

merkki `'.'` esittää vapaata käytävätilaa – siitä voi kulkea

merkki `'#'` esittää seinää – siitä ei voi kulkea

merkki `'*'` esittää voimapilleriä – sen voi syödä siitä kulkiessaan.

Nämä merkit piirtävät $N \times N$ ruudun kokoisen sokkelon nimeltä S kuvan 2 mukaisesti. Syöte-tiedoston rivillä $i+1$ ja sarakkeella j oleva merkki siis kertoo ruudun $S[i, j]$ sisällön — tiedoston ensimmäinen riviin sisälsi sokkelon S koon N .

Pelaaja aloittaa paikasta $S[1, 1]$, joka sisältää ensimmäisen voimapillerin `'*'`, jonka pelaaja syö heti aluksi.

Pelaaja voi astua nykyisestä paikastaan $S[i, j]$ yhdellä askeleella naapuripaikkaan

pohjoiseen $S[i-1, j]$ jos $i > 1$

etelään $S[i+1, j]$ jos $i < N$

länteen $S[i, j-1]$ jos $j > 1$

itään $S[i, j+1]$ jos $j < N$

```

123456789
+-----+
1|*.*##..|
2|.##..#..|
3|*##...#..|
4|.....###.|
5|###.*..#.|
6|..####.#.|
7|.....#.#|
8|####*....|
9|#####*|
+-----+

```

Kuva 2: Eräs Pac-Man-sokkelo jonka koko on $N = 9$. Kuvion reunus ja numerot eivät kuulu syötetiedostoon `pacman.in`: ne vain auttavat hahmottamaan reunuksen sisällä olevien merkkien rivit ja sarakkeet. Nämä reunuksen sisällä olevat merkit kuuluvat syötetiedostoon.

ja ellei tässä naapuripaikassa ole muuria '#'.

Pelaajan tavoitteena on päästä paikkaan $S[N, N]$. Siellä häntä odottaa palkinnoksi vielä yksi voimapilleri '*'.

Pelaaja syö voimapillerin $S[i, j] = '*'$ astumalla siihen paikkaan.

2.3 Tulostus

Ohjelmasi tulostaa tekstitiedostoon nimeltä `pacman.out`.

Tulostuksen pitää koostua yhdestä kokonaisluvusta p , joka on pienin sellainen luku, että paikasta $S[1, 1]$ on sellainen askeljono, joka vie paikkaan $S[N, N]$, ja jossa kahden peräkkäisen voimapillerin välillä kuluu korkeintaan p askelta.

Jos tällaista askeljonoa ei ole olemassa, niin vastaukseksi pitää silloin tulostaa $p = 0$.

Esimerkiksi kuvan 2 sokkelossa oikea vastaus on $p = 7$ askelta: jos voimapillerin vaikutus kestäisi yhtään lyhyemmän ajan, niin sokkelon keskellä olevasta voimapilleristä ei ehtisikään seuraavalle.

2.4 Pisteytys

Ohjelmaasi testataan 20 syötteellä. Näistä 10 syötteessä N on korkeintaan 30.

Ohjelmasi saa yhdelle syötteelle 64 megatavua työmuistia ja 1 sekunnin keskusyksikköaikaa.

Ohjelmasi saa yhdestä syötteestä

5 pistettä jos sen suoritus päättyy normaalisti ylittämättä muisti- tai aikarajoja, ja tulostaa tälle syötteelle oikean vastauksen

2 pistettä jos sen suoritus päättyy normaalisti ylittämättä muisti- tai aikarajoja, mutta sen tulostama vastaus on liian suuri

0 pistettä muuten.

Ohjelma saa kuitenkin pisteet oikeista vastauksistaan 0 vain jos se onnistuu lisäksi saamaan täydet 5 pistettä vähintään kahdesta (2) sellaisesta testistä, jossa oikea vastaus on suurempi kuin 0.

Työmuistiraja sisältää kaiken muun ohjelmasi käyttämän muistin paitsi sen suorituskelpoista ohjelmakoodia.

Keskusyksikköaikaraja sisältää sen ajan, jonka tietokoneen keskusyksikkö käyttää suorittaessaan sinun omaa ohjelmakoodiasi. Se ei sisällä sitä aikaa, jonka se käyttää suorittaessaan käyttöjärjestelmäpalveluita. Erityisesti se ei sisällä sitä aikaa, joka kuluu syötteen lukemiseen tiedostosta `pacman.in` työmuistiin tai valmiin vastauksen tulostamiseen tiedostoon `pacman.out`. Toisaalta se sisältää erilaisissa kirjastokutsuissa käytetyn ajan, esimerkiksi luetun syötteen muuntamisen tekstimuodosta vastaavaksi numeroksi.

Ohjelmat testataan sellaisella tietokoneella, jossa keskusyksikkönä on sellainen Intel Pentium P4 -mikroprosessori, jonka kellotaajuus on 3 GHz. Sen perusteella voit yrittää arvioida, kuinka pitkä vastaava keskusyksikköaikaraja on omalla koneellasi. Testikoneen käyttöjärjestelmänä on Linux.

2.5 Lähdekooditiedosto

Kirjoita ratkaisusi yhteen ASCII-lähdekooditiedostoon, jonka nimi riippuu käyttämästäsi ohjelmointikielestä seuraavasti:

KIELI	NIMI
C	<code>pacman.c</code>
C++	<code>pacman.cpp</code>
Pascal	<code>pacman.pas</code>

Laita tiedoston alkuun kommentti, jossa on seuraavat asiat:

- Teksti ”Datatähti-alkukilpailu 2005”.
- Oma nimesi ja syntymäaikasi.
- Tämän tehtävän nimi ”pacman”.
- Ilmoita, kääntyykö ohjelmasi kääntäjän oletusvalitsimilla. Jos ei, niin ilmoita lisäksi, mitkä valitsimet on lisättävä kääntäjän komentoriville. Ilmoita vain sellaiset valitsimet, joita ilman ohjelmasi ei käänny!

Tällainen välttämätön lisävalitsin voisi olla esimerkiksi matematiikkakirjastoa käyttävissä C-kielisissä ohjelmissa tarvittava ”-lm”.

Muuten tiedoston kirjoitusasu on vapaa. Älä kuitenkaan käytä yli 80 merkkiä pitkiä rivejä!

Ohjelmointikielillä C ja C++ kirjoitetut lähdekooditiedostot käännetään kääntäjällä GCC <<http://gcc.gnu.org/>>. Ohjelmointikielellä Pascal kirjoitetut tiedostot käännetään puolestaan kääntäjällä Free Pascal <<http://www.freepascal.org/>>, jonka pitäisi olla Turbo Pascal -yhteensopiva. Käytä kuitenkin vain valitsemasi ohjelmointikielen peruspiirteitä yhteensopivuuden takaamiseksi.

Käännökset tehdään kääntäjien oletusarvoisilla valitsimilla, ellet ole ilmoittanut muuta tiedoston alkukomentissa.

Huomioi vielä seuraavat tekniset vaatimukset:

- Ohjelmasi saa lukea syötettä vain tiedostosta nimeltä `pacman.in` ja tulostaa vain tiedostoon nimeltä `pacman.out`!

Erityisesti se *ei* saa lukea mitään näppäimistöltä tai tulostaa mitään näytölle.

- Kun ohjelmasi suoritus päättyy, niin sen täytyy palauttaa käyttöjärjestelmälle poistumisarvonaan 0. Muuten käyttöjärjestelmä katsoo, ettei ohjelmasi päättynytäkään normaalisti vaan johonkin virheeseen.

Ohjelmointikielissä C ja C++ tämä tarkoittaa, että pääohjelman `main` (ja koko ohjelman) suorituksen pitää päättyä komenttoon `return 0;` (eikä vain "valua" viimeiseen sulkevaan aaltosulkuun `}` saakka).

Ohjelmointikielessä Pascal kääntäjä huolehtii tästä automaattisesti.

Näiden vaatimusten rikkomisesta saattaa seurata pistemenetyksiä testiajojen epäonnistuesssa!

3 Ratkaisujen palautus

Tehtävien 1 ja 2 ratkaisujina palautetaan vastaavat lähdekooditiedostot. Suosittelen käyttämään sähköpostia. Myös kirjepostia voi käyttää, jos on pakko.

Jos sinulla on kysyttävää näistä tehtävistä tai muusta kilpailuun liittyvästä, niin ota yhteyttä sähköpostitse. Myös kirjepostia voi käyttää, jos on pakko, mutta silloin vastauksen saaminen vie kauemmin. Osoitteet annetaan alla.

Kilpailijoiden esittämät kysymykset ja vastaukset niihin tulevat kaikkien kilpailijoiden luettaviksi WWW-sivuston `<http://www.cs.helsinki.fi/matti.nykanen/datatahti/>` kautta. Tällä sivustolla voidaan tiedottaa muustakin kilpailuun liittyvästä, joten seuraa sitä!

Kysymysten pitäisi saapua perille viimeistään torstaina 11.11.2004, jotta niihin ehditään vastata hyvissä ajoin ennen kilpailuajan päättymistä.

3.1 Sähköpostilla

Palauta ratkaisusi sähköpostitse osoitteeseeni `matti.nykanen@cs.helsinki.fi`. Laadi viesti, jonka otsikko on ”Datatähti-ratkaisut 2005”, ja joka sisältää nimesi, syntymäaikasi ja yhteystietosi. Liitä sitten jokainen yllä mainituista lähdekooditiedostostasi omana MIME-liitteenään tähän viestiisi.

Sähköpostitse palautetut ratkaisut katsotaan saapuneiksi ajoissa, jos sähköpostin aikaleima on kilpailuajan sisällä. Tämä aikaleima tulee sähköpostin *välittäneestä* palvelinkoneesta, ei omasta mikrotietokoneestasi!

Jos MIME-liitteiden käyttö on mahdotonta, niin lähetä ensin yllä kuvattu viesti, jossa kerrot myös, että lähetät tiedostosi erikseen. Lähetä sen perään kukin lähdekooditiedostostasi omana viestinään. Tässä tapauksessa ratkaisujen katsotaan saapuneen ajoissa, jos *viimeisenkin* viestin aikaleima on kilpailuajan sisällä.

Voit lähettää myös kysymyksiäsi samaan sähköpostiosoitteeseen. Laita silloin viestisi otsikoksi ”Datatähti-kysymys 2005”. Kysymys ja vastaus ilmestyvät aikanaan yllä mainitulle verkkosivustolle; et siis saa erillistä vastausviestiä.

3.2 Kirjepostilla

Jos et voi käyttää sähköpostia, niin postita sen sijaan ratkaisusi seuraavaan osoitteeseen:

Matti Nykänen
Datatähti-alkukilpailu
Tietojenkäsittelytieteen laitos
PL 68 (Gustaf Hällströmin katu 2b)
00014 Helsingin yliopisto

Kirjeitse palautetut ratkaisut katsotaan saapuneiksi ajoissa, jos kirjekuoren postileima on kilpailuajan sisällä.

Kirjekuoren pitää sisältää seuraavat esineet:

- Yksi 3,5 tuuman DOS-formatoitu levyke, jonka juurihakemistoon olet tallettanut kyseiset lähdekooditiedostot.

Talleta lisäksi samaan juurihakemistoon ASCII-tekstitiedosto nimeltä `vastaja.txt` johon olet kirjoittanut oman nimesi, syntymäaikasi ja yhteystietosi sekä tiedon, että kyseessä on osallistuminen Datatähti-alkukilpailuun.

- Mielellään toinen samanlainen levyke varmuuden vuoksi.

(Ensimmäinen levyke voi kärsiä matkalla!)

- Tiedosto `vastaaja.txt` paperille tulostettuna.

(Siltä varalta että *molemmat* levykkeet kärsivät matkalla...)

Pakkaa levykkeet huolellisesti (käyttäen esimerkiksi pehmustettua kirjekuorta).

Voit lähettää myös kirjallisia kysymyksiäsi tähän samaan osoitteeseen. Liitä kysymyksiisi osoite, johon vastauksen voi lähettää.

(Julkaisupäivä: 02.11.2004.)