

# Datatähti 2008 -kilpailu

## Ohjelmointitehtävä 1/2:

### Kaupan paikka

**HUOM:** Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 5)

Yhden tien maa on erikoinen valtio. Siellä on ainoastaan yksi tie, ja tien varrella on tasan kilometrin välein yksi kylä (mukaanlukien tien kummassakin päässä). Kyliä on yhteensä  $N$  kappaletta, jolloin tien pituus on  $N - 1$  kilometriä. Tie ei haaraudu, eikä maassa ole tien varrella olevien kylien lisäksi muuta asutusta. Kylät on nimetty numeroilla  $0, 1, \dots, N - 1$  järjestyksessä tien alusta loppuun. Kussakin kylässä asuu yhtä monta asukasta, mutta eri kylien asukkaiden keskitulot vaihtelevat.

Kalle Kauppias haluaa avata myyntiliikkeen johonkin Yhden tien maan kylään. Liikkeen tulisi sijaita sellaisessa kylässä, jossa kysyntä olisi mahdollisimman suuri. Kalle arvioi, että liike tulisi keräämään asiakkaita myös muista kylistä, mutta sitä vähemmän mitä kauempana kylä sijaitsee. Tarkemmin sanottuna odotetut vuotuisat asiakasmäärät ovat seuraavat: kylästä, joka sijaitsee  $x$  kilometrin päässä, odotetaan tulevan  $M - x$  asiakasta vuodessa. Näin ollen kylistä, jotka sijaitsevat vähintään  $M$  kilometrin päässä, ei odoteta tulevan yhtään asiakasta. Koska kylä  $j$  sijaitsee  $|i - j|$  kilometrin päässä kylästä  $i$ , odotettaisiin kylään  $i$  sijoitettuun liikkeeseen tulevan kylästä  $j$  kaikkiaan  $\max\{M - |i - j|, 0\}$  asiakasta vuodessa.

Lisäksi Kallella on käytössään koon  $N$  omaava taulukko  $C$ , joka sisältää tilastotietoa siitä, kuinka paljon kunkin kylän asukkaat käyttävät rahaa ostoksiinsa: kunkin kylästä  $j$  tulevan asiakkaan voidaan odottaa kuluttavan liikkeessä  $C[j]$  rahayksikköä. Kun tämä yhdistetään odotettujen asiakasmäärien kanssa, saadaan kylästä  $j$  tulevien asiakkaiden kylään  $i$  sijoitetussa liikkeessä vuoden aikana kuluttamalle rahamäärälle arvio  $Ostot_i(j) = \max\{M - |i - j|, 0\} \cdot C[j]$ .

Odotettu kokonaisymyynti, jos liike sijoitetaan kylään  $i$ , saadaan edellisen pohjalta laskemalla yhteen arvot  $Ostot_i(j)$  kaikkien sellaisten kylien  $j$  osalta, että kylä  $j$  sijaitsee korkeintaan  $M - 1$  kilometrin päässä kylästä  $i$ . Jos oletetaan, että  $Ostot_i(j) = 0$  kun kylä  $j$  ei ole olemassa<sup>†</sup>, niin tämä kokonaisymyynnin ilmoittava summa on muotoa  $\sum_{j=-M+1}^{M-1} Ostot_i(j) = Ostot_i(i - M + 1) + Ostot_i(i - M + 2) + Ostot_i(i - M + 3) + \dots + Ostot_i(i - 2) + Ostot_i(i - 1) + Ostot_i(i) + Ostot_i(i + 1) + Ostot_i(i + 2) + \dots + Ostot_i(i + M - 2) + Ostot_i(i + M - 1)$ .

0    1    2    3    4    5    6    7    8    9 =  $N - 1$   
•-----•-----•-----•-----•-----•-----•-----•-----•-----•

$C$	9	47	92	83	49	99	61	21	93	52
-----	---	----	----	----	----	----	----	----	----	----

Yllä on esimerkki  $N = 10$  kylää käsittävästä tiestä sekä kyliä vastaavasta taulukosta  $C$ . Olkoon esimerkissä arvioinnin parametrina  $M = 3$ . Tällöin esimerkiksi liikkeen sijoittaminen kylään 5 antaisi kokonaisymyynnin  $Ostot_5(3) + Ostot_5(4) + Ostot_5(5) + Ostot_5(6) + Ostot_5(7) = 83 + 2 \cdot 49 + 3 \cdot 99 + 2 \cdot 61 + 21 = 621$ . Kylä 8 tuottaisi myynnin  $Ostot_8(6) + Ostot_8(7) + Ostot_8(8) + Ostot_8(9) = 61 + 2 \cdot 21 + 3 \cdot 93 + 2 \cdot 52 = 486$ . Esimerkissä kaikkein paras sijoituspaikka olisi kylä 3, joka tuottaisi myynnin  $Ostot_3(1) + Ostot_3(2) + Ostot_3(3) + Ostot_3(4) + Ostot_3(5) = 47 + 2 \cdot 92 + 3 \cdot 83 + 2 \cdot 49 + 99 = 677$ .

Tehtävänäsi on toteuttaa ohjelma, joka auttaa Kalle Kauppiasta laskemaan, mihin kylään liike tulee sijoittaa, jotta liikkeen odotettu vuotuinen kokonaisymyynti on mahdollisimman suuri.

<sup>†</sup>Kun  $j < 0$  tai  $j > N - 1$ .

## Syöte

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

Tiedoston **kauppa.in** ensimmäinen rivi sisältää kaksi toisistaan yhdellä välilyönnillä erotettua kokonaislukua  $N$  ja  $M$ , tässä järjestyksessä. Arvoille pätee  $0 < N \leq 32000000$  ja  $0 < M \leq 32000$ . Arvo  $N$  kertoo kylien lukumäärän, ja  $M$  on asiakasmäärien arvioinnin parametri.

Seuraavat  $N$  riviä ilmoittavat taulukon  $C$  arvot  $C[j]$  järjestyksessä  $j = 0 \dots N - 1$ . Kukin arvo  $C[j]$  on omalla rivillään ilman ylimääräistä tyhjää tilaa oleva kokonaisluku, jolle pätee  $1 \leq C[j] \leq 4096$ .

## Tuloste

Ohjelmasi kirjoittaa laskemansa vastauksen uuteen ASCII-tekstitiedostoon nimeltä **kauppa.out**.

Vastaustiedoston ensimmäiselle ja ainoalle riville tulostetaan kokonaisluku  $i$ , joka kertoo sen kylän numeron, johon liike tulee sijoittaa, jotta saavutetaan Kalle Kauppiaan arviointimenetelmän perusteella suurin kokonaismyynti. Rivin lopussa tulee olla rivinvaihto.

Jos vastaus ei ole yksikäsitteinen eli useampi kuin yksi sijoituspaikka johtaisi yhtä suureen ja samalla suurimpaan kokonaismyyntiin, tulee vastaustiedostoon kirjoittaa näistä parhaista kylistä ensimmäisen eli numeroltaan pienimmän kylän numero.

## Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla **kauppa.c**, **kauppa.cpp**, **kauppa.pas** tai **Kauppa.java**, riippuen ohjelmointikielestä (katso ohjeet sivulla 5).

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 192 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 (vastaa edellisen sivun kuvaa)

kauppa.in:

10 3  
9  
47  
92  
83  
49  
99  
61  
21  
93  
52

kauppa.out:

3

### Esimerkki 2

kauppa.in:

9 4  
22  
15  
10  
8  
11  
7  
22  
8  
10

kauppa.out:

1

---

# Datatähti 2008 -kilpailu

## Ohjelmointitehtävä 2/2: Pisin palindromi

**HUOM:** Tutustuthan huolellisesti tehtävien sääntöihin ja palautusohjeisiin (sivu 5)

---

Palindromi tarkoittaa merkkijonoa, joka on sama niin alusta loppuun kuin lopusta alkuun luetuna. Yksi tunnettu suomenkielinen palindromi on “saippuakivikauppias”. Palindromeja ei ole tutkittu pelkästään ajanvietteen vuoksi. Esimerkiksi DNA-sekvenssien yhteydessä on löydetty sellaisia toiminnallisia osasekvenssejä, jotka ovat palindromeja. Tässä suhteessa mielenkiintoinen kysymys voisi olla se, että mikä on pisin palindromi, jonka jokin merkkijono (teksti, esim. ASCII-tiedosto tai DNA-sekvenssi) sisältää.

Palindromi voitaisiin määritellä tarkemmin vaikkapa seuraavasti:  $N$  merkkiä omaava merkkijono  $A$  on palindromi jos ja vain jos  $A[j] = A[N - 1 - j]$  aina kun  $0 \leq j \leq N - 1$ . Tämän tehtävän yhteydessä merkkijonoa kohdellaan merkeistä koostuvana taulukkona, jossa merkit ovat järjestyksessä alusta loppuun. Esimerkiksi  $A[0]$  viittaa ensimmäiseen ja  $A[N - 1]$  viimeiseen merkkiin.

Vastaavasti voidaan määritellä, että  $N$  merkkiä omaavan merkkijonon  $A$  kohdassa  $i$  esiintyy  $M$  merkin pituinen palindromi jos ja vain jos  $A[i + j] = A[i + M - 1 - j]$  aina kun  $0 \leq j \leq M - 1$ . Tässä täytyy luonnollisesti päteä ehto  $0 \leq i \leq N - M$ , jotta ei mennä merkkijonon  $A$  rajojen yli.

Merkkijonon  $A$  kussakin kohdassa  $i = 0, \dots, N - 1$  esiintyy vähintään 1 merkin pituinen palindromi, sillä yksittäinen merkki  $A[i]$  on palindromi.

Esimerkiksi 16-merkkiä pitkä merkkijono  $A = \text{“abcbabdbbababada”}$  sisältää mm. seuraavat palindromit:

- kohta 0: 5 merkin pituinen palindromi “abcba”
- kohta 3: 3 merkin pituinen palindromi “bab”
- kohta 9: 3 merkin pituinen palindromi “aba”
- kohta 9: 5 merkin pituinen palindromi “ababa”

Tämän merkkijonon pisin palindromi on 5 merkin pituinen, ja tällainen esiintyy kohdissa 0, 8 ja 9.

Tehtävänäsi on toteuttaa ohjelma, joka selvittää, mikä on pisin syötteenä saadun ASCII-merkeistä muodostuvan merkkijonon sisältämä palindromi.

### Syöte

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

Tiedoston **palindro.in** ensimmäinen rivi sisältää yhden kokonaisluvun, joka kertoo merkkijonon  $A$  pituuden  $N$ . Arvolle  $N$  pätee  $1 \leq N \leq 32000000$ .

Seuraava rivi sisältää merkkijonon  $A$ . Rivillä on  $N$  merkkiä, jotka ovat merkit  $A[j]$  järjestyksessä  $j = 0 \dots N - 1$ . Kukin merkki kuuluu tavanomaiseen englanninkieliseen aakkostoon eli on joko välin a-z tai A-Z kirjain. Merkkien koolla on merkitystä eli esim. ‘b’  $\neq$  ‘B’ ja siten merkkijono “Oho” ei ole palindromi. Rivin lopussa on lisäksi rivinvaihto, eikä rivillä ole ylimääräistä tyhjää tilaa.

## Tuloste

Ohjelmasi kirjoittaa haun tuloksen uuteen ASCII-tekstitiedostoon nimeltä **palindro.out**.

Tiedoston ensimmäiselle ja ainoalle riville tulostetaan kaksi toisistaan erotettua kokonaislukua:  $M$  ja  $i$ . Arvo  $M$  kertoo pisimmän syötemerkkijonon  $A$  sisältämän palindromin pituuden. Arvo  $i$  kertoo missä kohtaa merkkijonoa  $A$  kyseinen  $M$  merkin pituinen palindromi esiintyy (palindromin ensimmäisen merkin indeksi merkkijonossa  $A$ ). Rivin lopussa tulee olla rivinvaihto.

Jos vastaus ei ole yksikäsitteinen eli löytyy useita yhtä pitkiä ja samalla pisimpiä palindromeja, tulee vastauksessa antaa näistä pienimmän indeksin  $i$  omaava palindromi (= sitä vastaavat arvot  $M$  ja  $i$ , aivan kuten yksikäsitteisen vastauksen yhteydessä).

## Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla **palindro.c**, **palindro.cpp**, **palindro.pas** tai **Palindro.java**, riippuen ohjelmointikielestä (katso ohjeet sivulla 5).

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin aikaa ja 192 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** (vastaa edellisen sivun esimerkkiä)

palindro.in:

16  
abcbabdbbababada

palindro.out:

5 0

**Esimerkki 2**

palindro.in:

25  
Onkotassatokaapalindromia

palindro.out:

10 2

# 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++, Pascal ja Java.

## Lähdekooditiedostot

Lähdekooditiedoston nimen tiedostopäätte määräytyy seuraavasti:

kieli	tiedostopäätte
C	c
C++	cpp
Pascal	pas
Java	java

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

1. Otsikkoteksti "Datatähti 2008 -alkukilpailu"
2. Tehtävän numero
3. Oma nimesi
4. Oma syntymäaikasi

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 nimetyt 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), Java-kielissä Java Platform SE:n luokat).

## 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
Java	Sun Java SE 6	Update 3	-	-

## Resurssirajat

Tehtävien kohdalla todettiin, että ohjelma saa käyttää kunkin testiajon aikana vain tietyn määrän aikaa ja muistia.

C/C++- sekä Pascal -kielen kohdalla em. resurssirajat asetetaan käyttäen Linuxin `ulimit` komentoa. Esimerkiksi 1 sekunnin aika- ja 192 megatavun muistiraja asetetaan komennolla `'ulimit -t 1 -v 196608'`.

Javan kohdalla muistiraja asetetaan Sunin Java-virtuaalikoneen parametrien `Xms` (muistin minimimäärä) ja `Xmx` (muistin maksimimäärä) avulla. Aikaraja kontrolloidaan parametrin `Xprof` avulla, joka tulostaa monenlaista tietoa ohjelman suorituksesta. Aika tulkitaan virtuaalikoneen tulostuksessa esiintyvän, muotoa

```
Flat profile of x secs (y total ticks): main
```

olevan rivin pohjalta (ohjelman suoritusaika sekunteina = arvo  $x$ ). Esimerkiksi kooditiedostosta **Ohjelma.java** käännetty ohjelma ajetaan testauksessa komennolla `'java -Xms192M -Xmx192M -Xprof Ohjelma'`.

Voit arvioida testikoneen nopeutta sen pohjalta, että siinä on AMD Athlon64 3200+ CPU.

## Ohjelmointitehtävien vastausten palautus

Jos samalta henkilöltä tulee kilpailuaikana 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](mailto:heikki.hyyro@cs.uta.fi). Viestin rakenne on seuraava:

- Sähköpostin otsikkokenttään (subject) teksti “Datatähti 2008 -alkukilpailun vastaukset”.
- Viestin rungoksi omat yhteystietosi:
  1. Nimesi
  2. Syntymäaikasi
  3. Postiosoitteesi
  4. Puhelinnumerosi
  5. Sähköpostiosoitteesi
- Kumpikin lähdekooditiedosto laitetaan sähköpostin mukaan erillisenä liitetiedostona.
  - 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!