

---

# Datatähti 2007 -kilpailu

## Ohjelmointitehtävä 1/2:

### Boolean AND -tyyppinen lausehaku

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

---

Hakusanojen esiintymiin perustuvat tiedonhakupalvelut, kuten **Google**, mahdollistavat usein ns. **Boolean logiikan** käytön. Tällaisessa Boolean haussa on mahdollista tarkentaa hakusanojen esiintymävaatimuksia loogisten operaattoreiden, kuten AND, OR ja NOT, avulla. Monet hakupalvelut soveltavat oletuksena AND-operaattoria kaikkiin hakusanoihin: haku palauttaa dokumentit, jotka sisältävät kaikki annetut hakusanat. Laadi tietokoneohjelma, joka toteuttaa tämänkaltaisen AND-operaattoria soveltavan Boolean sanahaun, kun haun kohteina ovat annetun tekstitiedoston sisältämät lauseet.

#### Syöte

Ohjelmasi lukee syötteenä kaksi ASCII-tekstitiedostoa, joiden nimet ovat **query.in** ja **text.in**.

Tiedosto **query.in** sisältää yhden tai useamman hakusanan. Jos hakusanoja on vähintään kaksi, on hakusanat erotettu toisistaan yhdellä välilyönnillä. Kukin hakusana on vähintään yhden ja korkeintaan 50 merkin pituinen, ja hakusanoja on korkeintaan 300. Yksinkertaisuuden vuoksi kukin hakusana koostuu englanninkielessä käytetyistä aakkosista tai numeroista, jolloin sallittujen merkkien joukko on  $\{a,b,c,\dots,z,A,B,C,\dots,Z,0,1,2,\dots,9\}$ .

Tiedosto **text.in** sisältää yhden tai useamman lauseen. Jos lauseita on vähintään kaksi, on lauseet erotettu toisistaan yhdellä välilyönnillä. Kukin lause koostuu yhdestä tai useammasta sanasta sekä välittömästi lauseen viimeisen sanan perässä olevasta pisteestä (muita väli- tai erikoismerkkejä ei ole). Jos lauseessa on vähintään kaksi sanaa, on sanat erotettu toisistaan yhdellä välilyönnillä. Sanojen pituus- ja merkkirajoitukset ovat samat kuin hakusanojen kohdalla: 1-50 merkkiä, joista kukin on numero tai englanninkielen aakkoston kirjain. Niin lauseiden lukumäärälle kuin yksittäisen lauseen sisältämien sanojen lukumäärällekin ei ole omaa ylärajaa. Ainoa rajoite on, että tiedoston **text.in** koko on korkeintaan 10 000 000 tavua.

#### Tuloste

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

Kyseessä on Boolean AND -tyyppinen haku, joka tarkoittaa että lause täsmää haun kanssa jos ja vain jos se sisältää kaikki annetut hakusanat. Hakusanat voivat sisältää jonkin sanan useammin kuin kerran. Tällöin täsmäävän lauseen täytyy sisältää kyseinen sana vähintään yhtä monta kertaa. Sanojen pitää täsmätä kokonaan, mutta merkkien kokoa ei huomioda. Esimerkiksi hakusana "auto" ei täsmää sanassa "autokauppa", mutta hakusana "Koira" täsmää sanan "kOIRA" kanssa.

Haun tuloste koostuu täsmäävien lauseiden järjestysnumeroista. Jos täsmääviä lauseita on vähintään kaksi, erotetaan tulostetut järjestysnumerot toisistaan yhdellä välilyönnillä. Jos mikään lause ei täsmää, tulostetaan luku 0.

Esimerkiksi jos tekstitiedoston **text.in** ensimmäinen ja viides lause, ja ainoastaan ne, sisältävät kunkin tiedostossa **query.in** olevan sanan, kirjoitetaan tiedostoon **query.out** luvut 1 ja 5 niin, että niiden välissä on yksi välilyönti.

## Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla **booleand.x**, missä tiedostopäätte *x* vastaa käyttämääsi ohjelmointikieltä (esim. **booleand.c**, katso ohjeet sivulla 7).

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin keskusyksikköaikaa ja 16 megatavua keskusmuistia. Kunkin testin kohdalla oikea tulos antaa 5 pistettä, ja väärä 0 pistettä. Aika- tai muistirajan ylittäminen tuottaa tulosteen oikeellisuudesta riippumatta 0 pistettä. Tehtävän maksimipistemäärä on siis 100 pistettä.

Jos vähintään kaksi kilpailijaa päätyy samaan kokonaispistemäärään, päätetään kilpailijoiden järjestys lopputuloksissa vertailemalla ratkaisujen suoritusajoja (nopeampi on parempi).

## Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

Kukin esimerkki käyttää samaa tekstitiedostoa **text.in**, joka on annettu alla.

text.in:

Tilastokeskuksen verkkosivuilla voi tutustua monenlaiseen Suomea koskevaan tilastotietoon. Poimin esimerkiksi tilastotietoa keskiansioista. Naisten keskipalkka vuonna 2004 oli 2137 euroa kuukaudessa. Miesten keskipalkka oli 2655 euroa kuukaudessa. Vuoden 2004 keskipalkka kaikkien palkansaajien osalta oli 2397 euroa kuukaudessa. Vuonna 2005 kukin mainittu keskipalkka oli melkein 100 euroa korkeampi. Silloin naisten keskipalkka oli 2227 euroa ja miesten 2750 euroa kuukaudessa. Kaikkien palkansaajien keskipalkka vuonna 2005 oli 2492 euroa kuukaudessa.

### Esimerkkihakua 1

query.in:

Keskipalkka Euroa

query.out:

3 4 5 6 7 8

### Esimerkkihakua 2

query.in:

KESKIPALKKA 2005 euroa

query.out:

6 8

### Esimerkkihakua 3

query.in:

tilasto

query.out:

0

### Esimerkkihakua 4

query.in:

EUROA kuukaudessa Euroa

query.out:

7

---

# Datatähti 2007 -kilpailu

## Ohjelmointitehtävä 2/2:

### Minigolf

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

---

Tarkastellaan yksinkertaista pienois- eli minigolfin tietokoneversiota. Minigolf-rata kuvataan suorakaiteen muotoisena ruudukkona, ja kukin ruutu voi olla joko tyhjä, seinänpala, pallon alkuruutu, tai reikä (pallon tavoiteruutu). Lisäksi oletetaan, että ruudukko on kokonaan seinän ympäröimä.

Pelin tavoite on lyödä pallo alkuruudusta reikään (loppuruutuun) mahdollisimman pienellä lyöntimäärällä. Kukin lyönti voi olla pituudeltaan vähintään yksi ja korkeintaan  $k$  ruutua, missä  $k$  on syötteessä annettu kokonaislukuparametri.

Tehtäväsi on toteuttaa ohjelma, joka etsii parhaan tavan selvittää annettu minigolf-rata. Ratkaisu ei välttämättä ole yksikäsitteinen, vaan voi olla monta erilaista lyöntien kombinaatiota, jotka saavat pallon reikään samalla (ja minimaalisella) lyöntimäärällä.

### Syöte

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

Tiedoston **minigolf.in** ensimmäinen rivi sisältää yhden kokonaisluvun, joka kertoo lyönnin maksimipituuden  $k$ .

Seuraavat  $m$  riviä, joista kukin sisältää tasan  $n$  merkkiä (sekä rivinvaihdon), kuvaavat  $m$  riviä ja  $n$  saraketta omaavan minigolf-ruudukon. Rivien ja sarakkeiden lukumäärille pätee  $1 \leq m \leq 3500$  ja  $1 \leq n \leq 3500$ . Kukin ruudukon kuvaukseen sisältyvä merkki on jokin alla luetelluista vaihtoehtoista.

- Tyhjä: '-'
- Seinänpala: 'X'
- Pallon alkuruutu: 'B'
- Reikä (pallon tavoiteruutu): 'H'

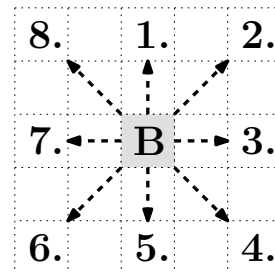
Luonnollisesti alkuruutu ja reikä ovat yksikäsitteisiä eli tiedostossa on täsmälleen yksi merkki 'B' sekä täsmälleen yksi merkki 'H'. Lisäksi voit olettaa, että annetulle ruudukolle on olemassa ratkaisu (eli pallo on mahdollista liikuttaa alkuruudusta loppuruutuun tyhjiä ruutuja pitkin). Alla on esimerkki tiedostosta **minigolf.in** (katso myös [Kuva 3](#)). Tässä lyönti voi olla korkeintaan viiden ruutuaskeleen pituinen, ja ruudukossa on 5 riviä sekä 8 saraketta.

```
5
-----
-----X--
H-X--X--
XXX----B
-----
```

### Lyönti

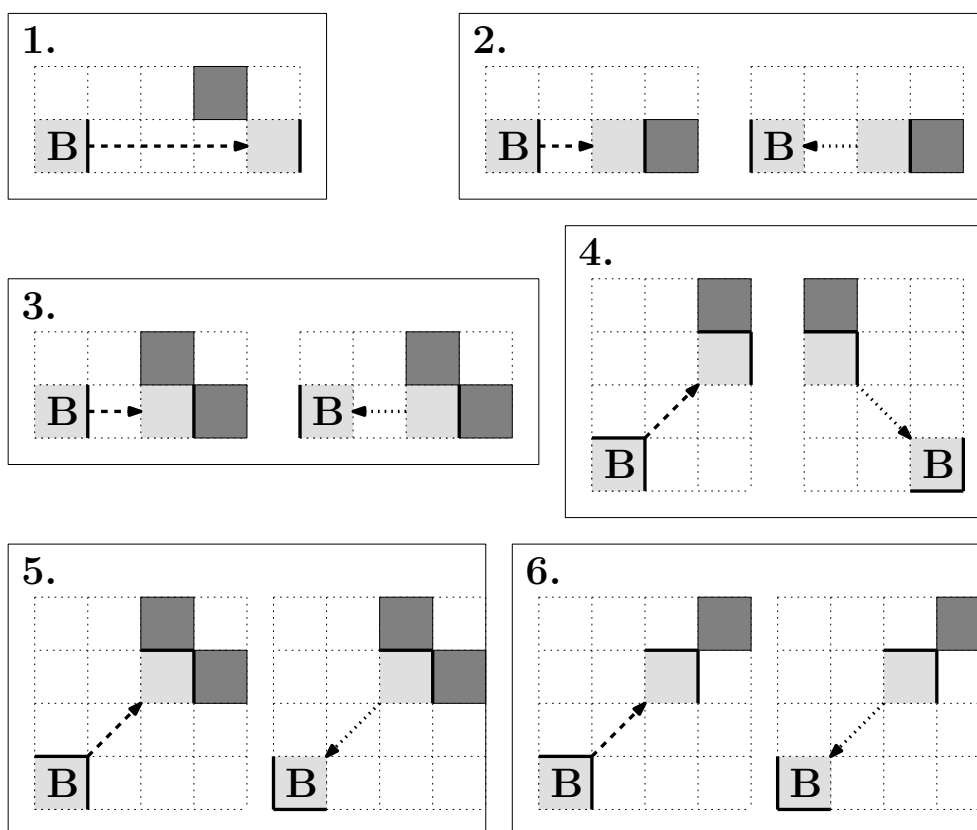
Yksittäinen lyönti on pituudeltaan vähintään 1 ja korkeintaan  $k$  ruutua, missä parametri  $k$  on annettu tiedoston **minigolf.in** ensimmäisellä rivillä. Lyönti voi suuntautua johonkin [Kuvassa 1](#) luetelluista kahdeksasta pääsuunnasta.

- |                           |                             |
|---------------------------|-----------------------------|
| 1. Ylös (YL)              | 5. Alas (AL)                |
| 2. Ylös oikealle 45° (YO) | 6. Alas vasemmalle 45° (AV) |
| 3. Oikealle (OI)          | 7. Vasemmalle (VA)          |
| 4. Alas oikealle 45° (AO) | 8. Ylös vasemmalle 45° (YV) |



Kuva 1: Pallon mahdolliset lyöntisuunnat.

Huomaa, että lyönnin pituutta mitataan *ruutuaskeleilla*. Esimerkiksi 3 askelta vasemmalle menevä lyönti katsotaan samanpituiseksi kuin 3 askelta vasemmalle yläviistoon menevä lyönti, vaikka euklidiseen etäisyyteen pohjautuen jälkimmäinen lyönti kulkeekin  $\sqrt{2}$ -kertaisen matkan.



Kuva 2: Pallon reagoiminen seinäkosketuksiin.

Pallo kimpoaa seinään osuessaan, mikäli osuma tapahtuu ennen kuin pallo on kulkenut lyönnin-pituuden määrittämän askelmäärän. Kimpoaminen ei vaikuta pallon kulkemien ruutuaskeleiden kokonaisuuteen, vaan se ainoastaan muuttaa pallon kulkusuuntaa. Pallon kimpoaminen noudattaa **Kuvassa 2** esitettyjä sääntöjä, jotka on selitetty alla. Pallon käyttäytymiseen vaikuttaa se, että se on pelissä tosiasiaa neliön muotoinen.

1. Pallon jokin muu kuin etenemissuunnan puoleinen sivu<sup>1</sup> koskettaa seinänpalaa<sup>2</sup>. Tällöin pallon suunta ei muutu.

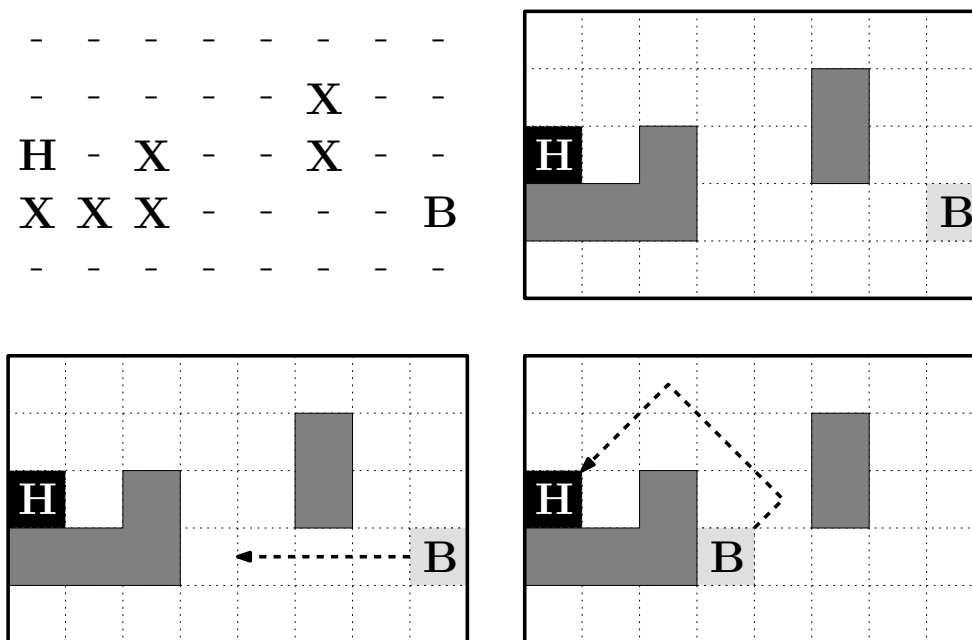
<sup>1</sup>Merkitty kuvassa paksunnetulla viivalla.

<sup>2</sup>Merkitty kuvassa tummanharmaalla.

2. Vaaka- tai pystysuoraan liikkuvan pallon etenemissuunnan puoleinen sivu törmää seinään. Tällöin pallo kimpoaa takaisin tulosuuntaansa.
3. Vaaka- tai pystysuoraan liikkuvan pallon etenemissuunnan puoleinen sivu törmää seinään, ja samalla myös toinen sivu koskettaa seinää. Pallo kimpoaa takaisin tulosuuntaansa.
4. 45 asteen kulmassa liikkuvan pallon toinen etenemissuunnasta näkyvä sivu osuu seinään. Pallo kimpoaa seinästä peilikuvana.
5. 45 asteen kulmassa liikkuvan pallon kumpikin etenemissuunnasta näkyvä sivu osuu seinään. Pallo kimpoaa seinästä takaisin tulosuuntaansa.
6. 45 asteen kulmassa liikkuvan pallon kumpikaan etenemissuunnasta näkyvä sivu ei osu seinään, mutta etenemissuunnan puoleinen kulma törmää seinän kulmaan. Pallo kimpoaa seinästä takaisin tulosuuntaansa.

Minigolfrata on selvitetty pallon siirtyessä reiän sijaintiruutuun (riippumatta siitä, olisiko kyseisessä lyönnissä vielä askeleita käyttämättä).

**Kuva 3** näyttää esimerkin pallon kimpoamisesta. Muista että ruudukkoa ympäröi seinä, joten ruudukon ulkoreunus<sup>3</sup> käyttäytyy kimpoamisen suhteen samalla tavalla kuin yhtenäinen seinäpaloista koostuva reunus.



Kuva 3: Esimerkki minigolfruudukosta ja sen ratkaisusta, kun  $k = 5$ . Vasemmalla ylhäällä on ruudukon tekstiesitys, oikealla ylhäällä on vastaava graafinen esitys, ja alarivin kuvat näyttävät tavan selvittää rata kahdella lyönnillä.

<sup>3</sup>Esitetty kuvassa paksunnetulla viivalla.

## Tuloste

Ohjelmasi kirjoittaa syötteenä annetun radan optimiratkaisun uuteen ASCII-tekstitiedostoon nimeltä **minigolf.out**.

Tiedoston ensimmäiselle riville tulostetaan yksi kokonaisluku  $h$ , joka kertoo minimimäärän lyöntejä, jolla annettu rata saadaan selvitettyä.

Seuraavat  $h$  riviä luettelevat järjestyksessä sellaiset  $h$  lyöntiä, jotka samassa järjestyksessä suoritettuna siirtävät pallon alkuruudusta reikään. Kukin näistä riveistä kuvaa yhden lyönnin. Yksittäisen lyönnin kuvaus koostuu lyönnin lähtösuunnan kaksikirjaimisesta lyhenteestä sekä lyönnin pituuden kertovasta kokonaisluvusta. Lyhenteen ja lyönnin pituuden välissä on yksi välilyönti. Lyöntisuuntien kaksikirjaimiset lyhenteet on esitetty suluissa [Kuvassa 1](#).

## Toteutustiedosto ja pisteytys

Ohjelmasi toteutuksen tulee olla yhdessä lähdekooditiedostossa. Tiedoston nimen tulee olla **minigolf.x**, missä tiedostopäätte  $x$  vastaa käyttämäsi ohjelmointikieltä (esim. **minigolf.c**, katso ohjeet sivulla [7](#)).

Ohjelmaasi testataan 20 kertaa erilaisilla syötteillä. Ohjelmasi saa kutakin testiä varten 1 sekunnin keskusyksikköaikaa ja 16 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 maksimipistemäärä on siis 100 pistettä.

Jos vähintään kaksi kilpailijaa päätyy samaan kokonaispistemäärään, päätetään kilpailijoiden järjestys lopputuloksissa vertailemalla ratkaisujen suoritusajoja (nopeampi on parempi).

## Esimerkkejä syötetiedostoista sekä vastaavista tulostiedostoista

### Esimerkkiruudukko 1 (vastaa [Kuvaa 3](#))

minigolf.in:

```
5
-----
-----X--
H-X--X--
XXX----B
-----
```

minigolf.out:

```
2
VA 4
YO 5
```

### Esimerkkiruudukko 2

minigolf.in:

```
6
B-X----X----
-----XXX
---X-XX----
XXXXX-X-----
-----X-XX--
--XX--X-----
-----
--X-----X-H-
```

minigolf.out:

```
3
AO 5
AL 5
AO 5
```

# Sääntöjä ja ohjeita

## Mitä palautetaan?

Kunkin ohjelmointitehtävien vastaus koostuu tehtävän yhteydessä mainitulla tavalla nimetystä yhdestä lähdekooditiedostosta. Siis kummallekin tehtävälle oma tiedosto. Älä lähetä esimerkiksi käännettyä 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 2007 -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 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`).
  - Ohjelmointikielessä C++ on käytössä myös kirjasto **Standard Template Library (STL)**, koska se on osa kielen standardia.

## Resurssirajat

Tehtävissä viitattiin seuraaviin resurssirajoihin:

Työmuisti	Käsittää kaiken muun ohjelmasi käyttämän muistin paitsi sen suorituskelpoista ohjelmakoodia.
Keskusyksikköaika	Sisältää sen ajan, jonka tietokoneen keskusyksikkö käyttää suorittaessaan sinun omaa ohjelmakoodiasi sekä sen kutsumia kirjastoaliohjelmia. Se ei sisällä sitä aikaa, jonka keskusyksikkö käyttää suorittaessaan käyttöjärjestelmäpalveluita, kuten tiedostojen avaamisia ja sulkemisia. Keskusyksikköajan mittarina toimii Linux/Unix ympäristön ns. "user time". Testitietokoneen nopeutta voit arvioida sen pohjalta, että siinä on AMD Athlon 3200+ -prosessori.

## Koodin ajoympäristö

Testitietokoneessa on Linux-käyttöjärjestelmä, ja kunkin ohjelmointikielen kanssa käytetään seuraavia kääntäjiä:

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

## Ohjelmointitehtävien vastausten palautus

Jos samalta henkilöltä tulee kilpailuaikana useita eri ratkaisuja, niin viimeinen niistä arvostellan.

Jos kilpailija lähettää vastauksensa myöhästyneenä, niin järjestäjillä on oikeus jättää se huomiotta.

Ratkaisujen palautus tapahtuu sähköpostitse osoitteeseen [heikki.hyyro@cs.uta.fi](mailto:heikki.hyyro@cs.uta.fi). Viestin rakenne on seuraava:

- Kirjoita sähköpostin otsikkokenttään (subject) teksti "Datatähti 2007 -alkukilpailun vastaukset".
- Kirjoita viestin rungoksi omat yhteystietosi:
  1. Nimesi
  2. Syntymäaikasi
  3. Postiosoitteesi
  4. Puhelinnumerosi
  5. Sähköpostiosoitteesi
- Kumpikin lähdekooditiedosto laitetaan sähköpostin mukaan erillisenä liitetiedostona.
  - Sähköpostiohjelmassa 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 näkyviin kisasivustolle ilman kysyjän nimeä.