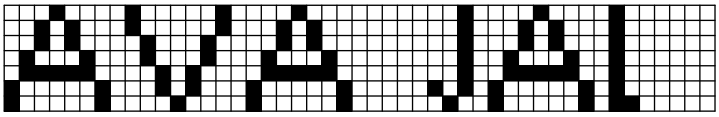
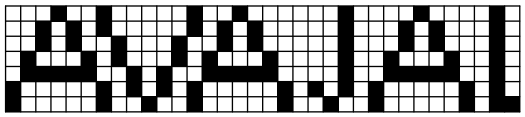
1. Koondamine

1 sekund 20 punkti

Teksti väljastamiseks rasterseadmele (ekraan, printer j.n.e.) esitatakse tähestiku iga täht $K \times L$ pikslist koosneva pildikesena. Näiteks alloleval joonisel on mõnede tähtede rasteresitustest moodustatud kujutis nii, et iga kahe naabertähe 7×7 rasteresituse vahele on lisatud üks tühi piksliveerg.



Selline naiivselt koostatud tekst ei näe enamasti kuigi hea välja, sest tähtede vahed on ebaühtlased. Näiteks eeltoodud joonisel on 'A' ja 'J' vahe suurem kui 'A' ja 'V' vahe, mis omakorda paistab suurem kui 'A' ja 'L' vahe. Mulje parandamiseks lükatakse uue tähe lisamisel rea lõppu seda vasakule nii palju kui võimalik, et ükski tema must piksel ei puutuks (ei serva- ega nurkapidi) kokku juba olemasolevate mustade pikslitega. Näitena toodud tekst omandab koondamisega alloleval joonisel toodud kuju.



Teksti paigutamiseks lehele on vaja teada iga sõna laiust pikslites. Kirjutada programm, mis saab tähestiku tähtede rasteresitused ja leiab antud teksti koondatud kuju laiuse pikslites.

Sisend. Selles ülesandes on sisendandmed kahes failis.

Tekstifailis `kernfont.sis` on 26 plokki, mis kirjeldavad ladina tähestiku tähti A…Z. Iga plokk koosneb 7 reast ja igal real on 7 täisarvu, kus 1 tähistab musta ja 0 valget pikslit. See fail on kõigis testides sama ja saadaval koos teiste sisendi ja väljundi näitefailidega võistluse serveris.

Tekstifaili `kern.sis` ainsal real on suurtest ladina tähtedest koosnev (tühikuteta) sõne S pikkusega kuni 100 märki. See fail on igas testis erinev.

Väljund. Tekstifaili `kern.val` ainsale reale väljastada sisendis antud sõne koondatud rasteresituse laius pikslites.

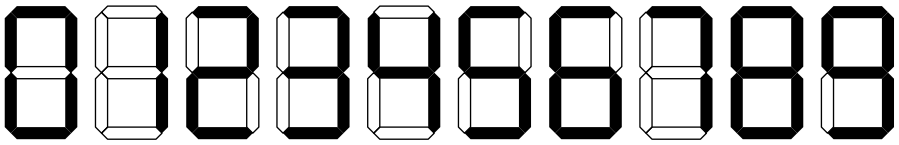
Näide.

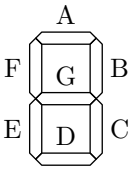
`kern.sis`	`kern.val`
`AVAJAL`	`34`

2. Kell

1 sekund 40 punkti

Digitaalkella tablool moodustatakse kellaaegade 00 : 00 kuni 23 : 59 näitamiseks numbrid 0...9 seitsme indikaatorsegmendi erinevate kombinatsioonide sisselülitamisega, nagu näha alloleval joonisel vasakul. Segmente tähistatakse kokkuleppeliselt A...G, nagu näidatud joonisel paremal.





Kui mõni segment läbi põleb, siis võib tablool paista midagi, mis pole üldse number. Näiteks alloleval joonisel vasakul on tunninäidu üheliste positsioonis ilmselt mõeldud näidata numbrit 8, kuid segment A ei tööta. Samuti võib juhtuda, et tablool paistev number pole see, mida on mõeldud. Näiteks minutinäidu üheliste positsioonis paistab number 6, kuid võib ka olla, et mõeldud on 8, aga segment B ei tööta.

Ka võib juhtuda, et kella muu elektroonika on rikkis ja tablool on näha midagi, mis ei saa üldse kellaaeg olla, isegi kui arvestada võimalusega, et tablool on mõni segment läbi põlenud. Selline olukord on näiteks ülaloleval joonisel paremal, kus tunninäidu kümneliste positsioonis saaks olla ainult number 8, mida aga selles kohas ühelgi kellaajal ei ole.

Kirjutada programm, mis leiab kõige varasema ja kõige hilisema kellaaja, mida võib olla mõeldud tablool näidata, kui on teada, millised segmendid parajasti põlevad ja millised ei põle.

Sisend. Tekstifailis `kell.sis` on neli rida, vastavalt tunninäidu kümneliste ja üheliste ning minutinäidu kümneliste ja üheliste positsiooni kirjeldus. Igal real on seitse tühikutega eraldatud täisarvu A, B, C, D, E, F, G. Igaüks neist arvudest võib olla kas 1 või 0, kus 1 tähendab, et vastav segment tablool põleb, ja 0, et ei põle — see kas pole sisse lülitatud või on rikkis.

Väljund. Tekstifaili `kell.val` esimesele reale väljastada varaseim ja teisele reale hiliseim võimalik kellaaeg kujul $HH : MM$, kus HH on kahekohaline tunni- ja MM kahekohaline minutinäit. Kui tabloo näit ei saa olla tekkinud ainult indikaatorsegmentide läbipõlemisega (s.t. ka muu elektroonika peab rikkis olema), väljastada faili ainsale reale sõna `VIGA`.

Näide.

`kell.sis`
```
1 1 1 1 1 1 0
0 1 1 1 1 1 1
1 1 1 1 1 1 0
1 0 1 1 1 1 1
```

`kell.val`
```
08:06
08:08
```

Näide.

`kell.sis`
```
0 1 0 0 1 1 1
0 1 1 1 1 1 1
1 1 1 1 1 1 0
1 0 1 1 1 1 1
```

`kell.val`
```
VIGA
```

Hindamine. Selles ülesandes saavad `VIGA`-vastusega testide eest punkte ainult need programmid, mis lahendavad õigesti vähemalt ühe teistsuguse vastusega testi.

3. Vead

1 sekund 40 punkti

Tarkvarafirmal on andmebaas vigadest, mille testijad või kasutajad on nende toodetest leidnud. Sageli pole vead omavahel sõltumatud, vaid ühe vea parandamise eelduseks on, et mingid teised vead oleks enne juba parandatud.

Lisaks juhtub vahel, et ühe ja sama vea kohta tuleb teateid erinevatest allikatest ja mõnikord ei pane vea andmebaasi sisestaja tähele, et selle vea kohta on juba kirje olemas. Kui sellist apsu hiljem märgatakse, ei saa duplikaati lihtsalt kustutada, sest kliendile on juba antud selle registreerimisnumber ja klient kasutab seda numbrit, et andmebaasist vaadata, kuidas tema mure lahendamine edeneb. Siis pannakse duplikaatkirje juurde viide, mis osutab teisele sama vea kohta käivale kirjele.

Kirjutada programm, mis leiab, mitu unikaalset viga on andmebaasis registreeritud ja mis järjekorras peaks neid parandama, et ühegi vea kallale ei asutaks enne, kui kõik selle parandamiseks vajalikud eeldused on täidetud.

Enne kirjete dubleerimise avastamist võib vigade vaheliste sõltuvuste andmeid olla andmebaasi sisestatud mitme ühe vea kohta käiva kirje juurde. Tööde järjekorra planeerimisel tuleb arvestada ka kõigi duplikaatkirjete juurde märgitud sõltuvustega.

Sisend. Tekstifaili `vead.sis` esimesel real on andmebaasi kirjete koguarv N ($1 \leqslant N \leqslant 50\,000$), duplikaadiviidete arv K ($1 \leqslant K \leqslant 1000$) ja vigade vaheliste sõltuvuste arv M ($1 \leqslant M \leqslant 50\,000$). Kirjed andmebaasis on nummerdatud $1 \ldots N$.

Faili järgmisel K real on igaühel kaks täisarvu A_i ja B_i ($1 \leqslant A_i \leqslant N$, $1 \leqslant B_i \leqslant N$), mis näitavad, et kirje B_i on tegelikult kirje A_i duplikaat.

Faili järgmisel M real on igaühel kaks täisarvu C_i ja D_i ($1 \leqslant C_i \leqslant N$, $1 \leqslant D_i \leqslant N$), mis näitavad, et vea D_i parandamiseks peab viga C_i olema juba parandatud.

Väljund. Tekstifaili `vead.val` esimesele reale väljastada unikaalsete vigade arv V ja järgmisele V reale vigade kirjete numbrid selles järjekorras, nagu neid vigu saab parandama hakata. Kirjete numbrid väljastada igaüks eraldi reale. Kui võimalikke järjekordi on mitu, väljastada ükskõik milline neist. Võib eeldada, et leidub vähemalt üks järjekord. Igast duplikaatkirjete komplektist võib väljastada ükskõik millise kirje numbri.

Näide.

`vead.sis`

```
7 3 3
7 6
6 1
2 5
2 1
5 4
3 4
```

`vead.val`

```
4
2
3
1
4
```

Kirjed 1, 6 ja 7 on kõik üksteise duplikaadid. Kirjed 2 ja 5 on ka duplikaadid. Vea 4 parandamine sõltub vigade 3 ja 2-5 parandamisest. Samuti sõltub vea 1-6-7 parandamine vea 2-5 parandamisest. Vigade 1-6-7 ja 4, samuti 1-6-7 ja 3 ning 2-5 ja 3 omavaheline järjekord võib olla ükskõik milline.

Hindamine. Selles ülesandes annab unikaalsete vigade arvu leidmine 25% punktidest. Kui teie lahendus vigade parandamise järjekorda ei leia, jätke see lihtsalt väljastamata (tehke ühe reaga väljundfail).

1. Telefoniraamat

1 sekund 20 punkti

Vaatleme järgmist eeskirja telefoniraamatust nime järgi telefoninumbri otsimiseks:

- vaatame raamatu keskmisele leheküljele (kui lehekülgi on paarisarv, valime raamatu alguse pool oleva);
- kui otsitav nimi on sellel leheküljel, lõpetame otsimise;
- kui otsitav nimi on tähestikus eespool, jätame kõrvale vaadeldava lehekülje ja kõik sellele järgnevad ning otsime sama eeskirja järgi raamatu allesjäänud osas;
- kui otsitav nimi on tähestikus tagapool, jätame kõrvale vaadeldava lehekülje ja kõik sellele eelnevad ning otsime sama eeskirja järgi raamatu allesjäänud osas.

Kirjutada programm, mis leiab telefoniraamatu paksuse ja otsitava nime asukoha järgi, milliseid raamatu lehekülgi otsingu käigus vaadatakse.

Sisend. Tekstifaili `tele.sis` ainsal real on kaks tühikuga eraldatud täisarvu, telefoniraamatu lehekülgede arv N ($1 \leqslant N \leqslant 1\,000\,000$) ja lehekülje number L ($1 \leqslant L \leqslant N$), millel otsitav nimi asub.

Väljund. Tekstifaili `tele.val` väljastada nende lehekülgede numbrid, mida otsingu käigus vaadeldakse. Numbrid väljastada lehekülgede vaatamise järjekorras, igaüks eraldi reale.

Näide.

`tele.sis`
```
9 3
```

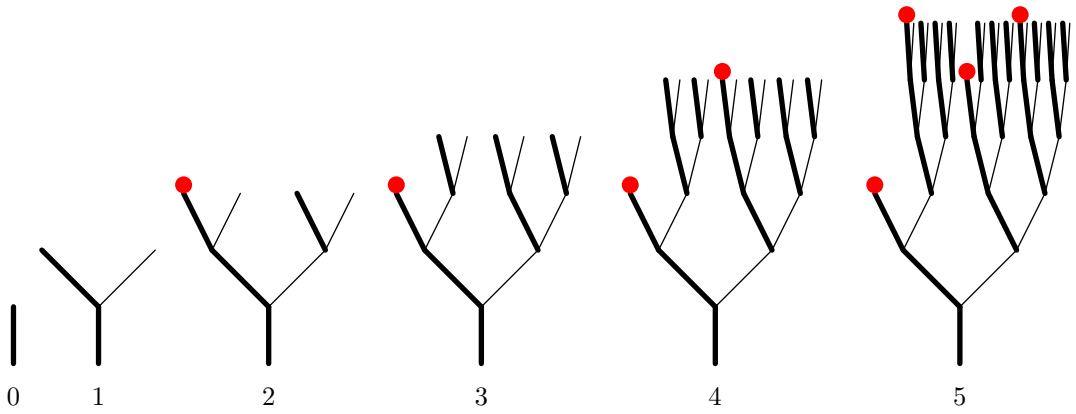
`tele.val`
```
5
2
3
```

9-leheküljelises telefoniraamatus alustame otsingut 5. leheküljelt. Sinna vaadates näeme, et vajalik kanne peab olema raamatu alguse pool, seega jätkame otsimist lehekülgedel 1 kuni 4. Edasi vaatame 2. leheküljele ja näeme, et vajalik kanne peab olema raamatu lõpu pool, seega jätkame otsimist lehekülgedel 3 kuni 4. Siis vaatame 3. leheküljele, kust leiamegi otsitavad andmed.

2. Taim

1 sekund 40 punkti

Toataim *planta progressibilia* kasvab kindlate reeglite järgi. Igal nädalal hargneb taime iga võrse kaheks, peaharuks ja kõrvalharuks, mis mõlemad kasvavad nädalaga ühe ühiku võrra pikemaks. Kui taime mingi haru on peaharuna kasvanud kolm nädalat, puhkeb kolmanda nädala lõpus tema tipus kaunis õis ja see haru enam edasi ei kasva.



Kirjutada programm, mis leiab, mitu uut õit puhkeb taimel N-ndal nädalal pärast seda, kui potti istutatakse ühe ühiku pikkune peaharu.

Sisend. Tekstifaili `taim.sis` ainsal real on uuritava nädala järjekorranumber N ($1 \leqslant N \leqslant 75$).

Väljund. Tekstifaili `taim.val` ainsale reale väljastada N nädalat pärast istutamist puhkevate uute õite arv.

Näide.

`taim.sis`	`taim.val`
`5`	`2`

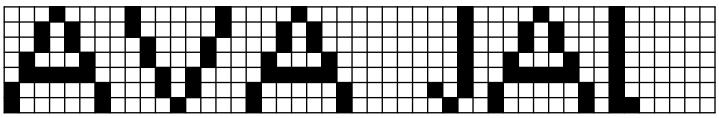
Näide.

`taim.sis`	`taim.val`
`6`	`3`

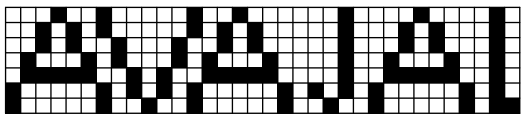
3. Koondamine

1 sekund 40 punkti

Teksti väljastamiseks rasterseadmele (ekraan, printer j.n.e.) esitatakse tähestiku iga täht $K \times L$ pikslist koosneva pildikesena. Näiteks alloleval joonisel on mõnede tähtede rasteresitustest moodustatud kujutis nii, et iga kahe naabertähe 7×7 rasteresituse vahele on lisatud üks tühi piksliveerg.



Selline naiivselt koostatud tekst ei näe enamasti kuigi hea välja, sest tähtede vahed on ebaühtlased. Näiteks eeltoodud joonisel on 'A' ja 'J' vahe suurem kui 'A' ja 'V' vahe, mis omakorda paistab suurem kui 'A' ja 'L' vahe. Mulje parandamiseks lükatakse uue tähe lisamisel rea lõppu seda vasakule nii palju kui võimalik, et ükski tema must piksel ei puutuks (ei serva- ega nurkapidi) kokku juba olemasolevate mustade pikslitega. Näitena toodud tekst omandab koondamisega alloleval joonisel toodud kuju.



Teksti paigutamiseks lehele on vaja teada iga sõna laiust pikslites. Kirjutada programm, mis saab tähestiku tähtede rasteresitused ja leiab antud teksti koondatud kuju laiuse pikslites.

Sisend. Selles ülesandes on sisendandmed kahes failis.

Tekstifailis `kernfont.sis` on 26 plokki, mis kirjeldavad ladina tähestiku tähti A. . . Z. Iga plokk koosneb 7 reast ja igal real on 7 täisarvu, kus 1 tähistab musta ja 0 valget pikslit. See fail on kõigis testides sama ja saadaval koos teiste sisendi ja väljundi näitefailidega võistluse serveris.

Tekstifaili `kern.sis` ainsal real on suurtest ladina tähtedest koosnev (tühikuteta) sõne S pikkusega kuni 100 märki. See fail on igas testis erinev.

Väljund. Tekstifaili `kern.val` ainsale reale väljastada sisendis antud sõne koondatud rasteresituse laius pikslites.

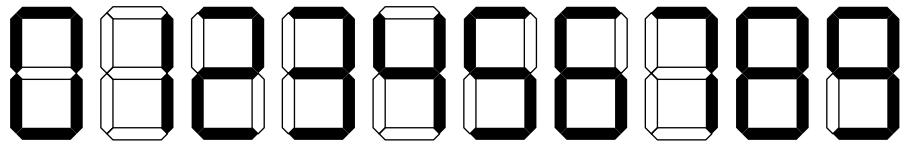
Näide.

`kern.sis`	`kern.val`
`AVAJAL`	`34`

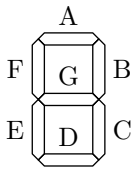
1. Numbrinäit

1 sekund 20 punkti

Paljudel tabloodel moodustatakse numbrid 0...9 seitsme indikaatorsegmendi erinevate kombinatsioonide sisselülitamisega, nagu näha alloleval joonisel.



Segmente tähistatakse kokkuleppeliselt tähtedega A...G, nagu näidatud alloleval joonisel.



Kui mõni segment läbi põleb, siis võib tablool paista midagi, mis pole üldse number. Näiteks alloleval joonisel vasakul on ilmselt mõeldud näidata numbrit 8, kuid segment A ei tööta. Samuti võib juhtuda, et tablool paistev number pole see, mida on mõeldud. Näiteks alloleval joonisel paremal paistab number 6, kuid võib ka olla, et mõeldud on 8, aga segment B ei tööta.

Kirjutada programm, mis leiab kõik numbrid, mida võib olla mõeldud tablool näidata, kui on teada, millised segmendid parajasti põlevad ja millised ei põle.

Sisend. Tekstifaili `nait.sis` ainsal real on seitse tühikutega eraldatud täisarvu A, B, C, D, E, F, G. Igaüks neist arvudest võib olla kas 1 või 0, kus 1 tähendab, et vastav segment tablool põleb, ja 0, et ei põle — see kas pole sisse lülitatud või on rikkis.

Väljund. Tekstifaili `nait.val` väljastada kõik numbrid, mida sisendis kirjeldatud tabloonäiduga võib mõeldud olla. Võimalikud numbrid väljastada kasvavas järjekorras, igaüks eraldi reale.

Näide.

`nait.sis`
```
0 1 1 1 1 1 1
```

`nait.val`
```
8
```

Näide.

`nait.sis`
```
1 0 1 1 1 1 1
```

`nait.val`
```
6
8
```

2. Tööplaan

1 sekund 40 punkti

Maja ehitamiseks tuleb teha ära hulk töid, aga mõne töö alustamiseks võib olla vaja, et mingid teised tööd oleks enne tehtud. Näiteks ei saa hakata katust panema enne kui seinad on püsti, aga seinu ei saa laduda enne kui vundament on valmis.

Kirjutada programm, mis saab andmed selle kohta, millised tööd millistest teistest sõltuvad ja väljastab nende tööde nimekirja, millega saab kohe alustada.

Sisend. Tekstifaili `tood.sis` esimesel real on kaks tühikuga eraldatud täisarvu, tööde arv N ($1 \leqslant N \leqslant 50\,000$) ja tööde vaheliste sõltuvuste arv M ($1 \leqslant M \leqslant 50\,000$). Tööd on nummerdatud $1 \ldots N$. Faili järgmisel M real on igaühel kaks tühikuga eraldatud täisarvu A_i ja B_i ($1 \leqslant A_i \leqslant N$, $1 \leqslant B_i \leqslant N$), mis näitavad, et töö B_i alustamiseks peab töö A_i olema lõpetatud.

Väljund. Tekstifaili `tood.val` väljastada nende tööde numbrid, mida saab kohe tegema hakata. Tööde numbrid väljastada igaüks eraldi reale. Numbrite järjekord failis pole oluline, aga iga number tuleb väljastada täpselt üks kord. Võib eeldada, et leidub vähemalt üks töö, mida saab kohe tegema hakata.

Näide.

`tood.sis`

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

`tood.val`

```
2
3
```

3. Telefoniraamat

1 sekund 40 punkti

Vaatleme järgmist eeskirja telefoniraamatust nime järgi telefoninumbri otsimiseks:

- vaatame raamatu keskmisele leheküljele (kui lehekülgi on paarisarv, valime raamatu alguse pool oleva);
- kui otsitav nimi on sellel leheküljel, lõpetame otsimise;
- kui otsitav nimi on tähestikus eespool, jätame kõrvale vaadeldava lehekülje ja kõik sellele järgnevad ning otsime sama eeskirja järgi raamatu allesjäänud osas;
- kui otsitav nimi on tähestikus tagapool, jätame kõrvale vaadeldava lehekülje ja kõik sellele eelnevad ning otsime sama eeskirja järgi raamatu allesjäänud osas.

Kirjutada programm, mis leiab telefoniraamatu paksuse ja otsitava nime asukoha järgi, milliseid raamatu lehekülgi otsingu käigus vaadatakse.

Sisend. Tekstifaili `tele.sis` ainsal real on kaks tühikuga eraldatud täisarvu, telefoniraamatu lehekülgede arv N ($1 \leqslant N \leqslant 1\,000\,000$) ja lehekülje number L ($1 \leqslant L \leqslant N$), millel otsitav nimi asub.

Väljund. Tekstifaili `tele.val` väljastada nende lehekülgede numbrid, mida otsingu käigus vaadeldakse. Numbrid väljastada lehekülgede vaatamise järjekorras, igaüks eraldi reale.

Näide.

`tele.sis`
```
9 3
```

`tele.val`
```
5
2
3
```

9-leheküljelises telefoniraamatus alustame otsingut 5. leheküljelt. Sinna vaadates näeme, et vajalik kanne peab olema raamatu alguse pool, seega jätkame otsimist lehekülgedel 1 kuni 4. Edasi vaatame 2. leheküljele ja näeme, et vajalik kanne peab olema raamatu lõpu pool, seega jätkame otsimist lehekülgedel 3 kuni 4. Siis vaatame 3. leheküljele, kust leiamegi otsitavad andmed.