1. Jada

1 sekund 10 punkti

Nimetame märgivahetusteks arvujadas kohti, kus positiivsele elemendile järgneb negatiivne või negatiivsele positiivne nii, et nende vahel kas pole üldse teisi elemente või on seal ainult nullid.

Kirjutada programm, mis leiab märgivahetuste arvu antud täisarvude jadas.

Sisend. Tekstifaili `jada.sis` esimesel real on jada elementide arv N ($1 \leqslant N \leqslant 1000$). Faili teisel real on N tühikutega eraldatud täisarvu absoluutväärtustega kuni 10 000 — jada elemendid.

Väljund. Tekstifaili `jada.val` ainsale reale väljastada üks täisarv — märgivahetuste arv sisendis kirjeldatud jadas.

Näide.

jada.sis	jada.val
3 -1 1 -1	2

Näide.

jada.sis	jada.val
3 -1 0 1	1

2. Kuupäev

1 sekund 20 punkti

Vaatleme kuupäevade esitusi 8-kohaliste arvudena kujul $AAAAKKPP$, kus $AAAA$ on (alati neljakohaline) aasta, KK (alati kahekohaline) kuu ja PP (alati kahekohaline) päeva number.

Nimetame palindroomilisteks selliseid kuupäevi, mille esitused on samad nii vasakult paremale kui paremalt vasakule lugedes. Näiteks 2. november 2011 (20111102) on palindroomiline, aga 2. detsember 2012 (20121202) ei ole.

Kirjutada programm, mis leiab lähima antud kuupäevale järgneva palindroomilise kuupäeva.

Sisend. Tekstifaili `kuup.sis` ainsal real on ühe kuupäeva esitus kujul $AAAAKKPP$.

Väljund. Tekstifaili `kuup.val` ainsale reale väljastada sisendis antud kuupäevale järgnevatest palindroomilistest kuupäevadest varaseima esitus kujul $AAAAKKPP$. Võib eeldada, et igas testis on selline (ülimalt neljakohalise aastanumbriga) kuupäev olemas.

Liigaastate arvestuses eeldada Gregoriuse kalendri kasutamist alates aastast 1 kuni aastani 9999. See tähendab, et liigaastateks lugeda need, mille numbrid jaguvad neljaga, kuid ei jagu sajaga, ja lisaks need, mille numbrid jaguvad neljasajaga.

Näide.

kuup.sis	kuup.val
20111101	20111102

Näide.

kuup.sis	kuup.val
20111102	20200202

3. Maatriks

1 sekund 30 punkti

3×3 maatriksi (arvutabeli)

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

determinandiks nimetatakse avaldist

$$(a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}) - (a_{31}a_{22}a_{13} + a_{32}a_{23}a_{11} + a_{33}a_{21}a_{12}).$$

Kirjutada programm, mis leiab, kui palju erinevaid determinandi absoluutväärtusi on võimalik saada antud 9 täisarvu 3×3 maatriksisse paigutamisel.

Sisend. Tekstifaili `maat.sis` ainsal real on 9 tühikutega eraldatud täisarvu, mille absoluutväärtused ei ületa 100.

Väljund. Tekstifaili `maat.val` ainsale reale väljastada sisendis antud arvude kõikvõimalikel viisidel maatriksisse paigutamisel saadavate erinevate determinandi absoluutväärtuste arv.

Näide.	`maat.sis`	`maat.val`
	`1 1 2 2 0 0 0 0 0`	`3`

Võimalikud determinandi väärtused on 4 (näiteks maatriks A_1 allolevas tabelis), 2 (näiteks A_2), 0 (näiteks A_3), -2 (näiteks A_4) ja -4 (näiteks A_5). Muid väärtusi ei ole võimalik saada, seega ongi erinevaid absoluutväärtusi vaid kolm: 0, 2 ja 4.

A_1	A_2	A_3	A_4	A_5
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 1 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 2 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 2 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

4. Vali mind!

1 sekund 40 punkti

Ühes liitriigis on kord, et presidendivalimised toimuvad kahe kandidaadi vahel kahes voorus. Esimeses voorus hääletavad kõik elanikud. Igal osariigil on kindel arv valijamehi, kes valimiste teises voorus hääletavad selle kandidaadi poolt, kes sai nende osariigis esimeses voorus rohkem hääli. Teises voorus võidab kandidaat, kes saab valijameestelt rohkem hääli.

Mõlemas voorus on võitmiseks vaja saada vastaskandidaadist rohkem hääli. Kui esimeses voorus jagunevad mõnes osariigis hääled kahe kandidaadi vahel võrdselt, siis selle osariigi valijamehed teises voorus ei hääleta. Viigi korral teises voorus tulevad kordusvalimised.

Tänavustel valimistel juhtus nii, et üks kandidaatidest otsustas hääli osta, aga see plaan tuli avalikuks. Nagu tavaliselt, ei suutnud asja ametlik uurimine midagi tõestada, nii et sellel kandidaadil valimistel osalemine otseselt keelatud ei ole. Küll aga hääletavad nüüd tema poolt ainult need valijad, kellele ta maksab; kõik teised hääletavad vastaskandidaadi poolt.

Kirjutada programm, mis leiab minimaalse hulga valijaid, kellele see kandidaat peab esimeses voorus maksma, et teises voorus valituks saada.

Sisend. Tekstifaili `vali.sis` esimesel real on osariikide arv N ($1 \leqslant N \leqslant 100$). Järgmisel N real on igaühel kaks tühikuga eraldatud täisarvu — ühe osariigi elanike arv M_i ($1 \leqslant M_i \leqslant 10^6$) ja valijameeste arv K_i ($1 \leqslant K_i \leqslant 10^3$, $K_i \leqslant M_i$). Osariigid on nummerdatud $1 \ldots N$ nende andmete sisendis esitamise järjekorras.

Väljund. Tekstifaili `vali.val` esimesele reale väljastada täisarv K — nende osariikide arv, kus tuleb elanikke ära osta. Järgmisele K reale väljastada igaühele kaks tühikuga eraldatud täisarvu — ühe osariigi number ja selles osariigis ära ostetavate valijate arv. Kui minimaalse ostetavate valijate arvuga lahendusi on mitu, väljastada ükskõik milline neist.

Näide.

vali.sis	vali.val
3	2
121 2	1 61
150 3	3 71
140 4	

Näide.

vali.sis	vali.val
3	2
120 2	1 60
150 3	3 71
140 4	

Hindamine. Testides summaarse väärtusega 20 punkti on $N < 15$.

5. Kivilinn

1 sekund 50 punkti

Linnapea tahab välja vahetada raekoja platsi katvad kiviplaadid. Kivimurrust on võimalik saada igasugustes mõõtudes ristkülikukujulisi plaate. Linnapea on ka perfektsionist ja tahab väljaku katta võimalikult väikese arvu plaatidega.

Kirjutada programm, mis loeb sisse olemasolevate põhja-lõuna ja ida-lääne suunaliste servadega ristkülikukujuliste plaatide andmed ja leiab minimaalse uute plaatide arvu, millega on võimalik täpselt sama ala täpselt ühes kihis katta.

Sisend. Tekstifaili `kivi.sis` esimesel real on olemasolevate plaatide arv N ($1 \leqslant N \leqslant 500$). Järgmisel N real on igaühel neli tühikutega eraldatud täisarvu — ühe plaadi põhja- ja lõunaserva kaugused platsi põhjaservast ning lääne- ja idaserva kaugused platsi lääneservast. Kõik kaugused on täisarvud $0 \ldots 10\,000$.

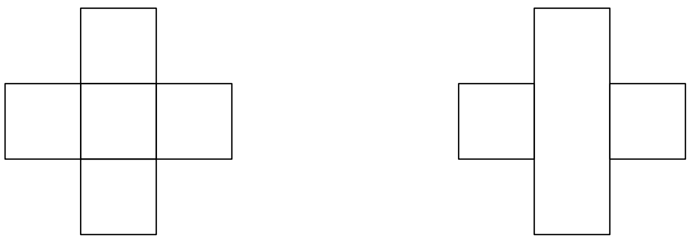
Võib eeldada, et plats on sidus, s.t. platsi igast punktist on võimalik minna platsi igasse teise punkti ilma platsilt väljumata. Lisaks on teada, et mistahes põhja-lõuna või ida-lääne suunaline sirge lõikub platsi äärega ülimalt kaks korda.

Väljund. Tekstifaili `kivi.val` esimesele reale väljastada minimaalne võimalik uute plaatide arv K ja järgmisele K reale igaühele neli tühikutega eraldatud täisarvu — ühe plaadi põhja- ja lõunaserva kaugused platsi põhjaservast ning lääne- ja idaserva kaugused platsi lääneservast. Kui minimaalse plaatide arvuga lahendusi on mitu, väljastada ükskõik milline neist.

Näide.

kivi.sis	kivi.val
5	3
20 30 10 20	20 30 10 20
10 20 20 30	10 40 20 30
20 30 20 30	20 30 30 40
30 40 20 30	
20 30 30 40	

Alloleval joonisel on vasakul näidatud vanade ja paremal uute plaatide paigutus.



Hindamine. Testides summaarse väärtusega 20 punkti on $N < 15$.

6. Elektritarbimine

Testimine 50 punkti

Kauglugemisega elektriarvesti teatab elektrifirmale iga tunni kohta kliendi keskmise tarbitud võimsuse selle tunni jooksul. Elektrifirma tellis tarkvarafirmalt programmi, mis peab ühe kuu tunnitarbimiste andmete põhjal arvutama kliendi päevase ja öise energiatarbimise summad selle kuu jooksul.

Vastavalt elektrifirma lepingutingimustele loetakse päevaseks tarbimiseks talveajal tööpäeviti kell 7:00 kuni 23:00 ja suveajal tööpäeviti 8:00 kuni 24:00 tarbitud energia; kogu ülejäänud tarbimine loetakse öiseks.

Suveaeg algab igal aastal märtsi viimasel pühapäeval, kui (sel hetkel kehtiva talveaja järgi) kell 3:00 keeratakse kell ühe tunni võrra edasi (see tähendab, et tundi 3:00–4:00 sel päeval ei ole) ja lõpeb oktoobri viimasel pühapäeval, kui (sel hetkel kehtiva suveaja järgi) kell 4:00 keeratakse kell ühe tunni võrra tagasi (see tähendab, et tundi 3:00–4:00 on sel päeval kaks korda, üks suve- ja teine talveaja järgi).

Programm peab õigesti arvestama ka liigaastaid, mis vastavalt kehtivale Gregoriuse kalendrile on need, mille numbrid jaguvad neljaga, kuid ei jagu sajaga, ja lisaks need, mille numbrid jaguvad neljasajaga.

Koostada elektrifirmale testandmete komplekt, millega enne tarkvarafirmale tasumist nende kirjutatud programmi õigsust kontrollida.

Sisend. Tekstifaili `tarb.sis` esimesel real on kaks tühikuga eraldatud täisarvu A ja K, kus A on analüüsitava perioodi aasta- ja K kuunumber ($1900 \leqslant A \leqslant 2100$, $1 \leqslant K \leqslant 12$). Faili teisel real on tarbimiskannete arv N ($0 \leqslant N \leqslant 745$). Faili järgmisel N real on igaühel kaks tühikuga eraldatud täisarvu T ja P, kus T on kandele vastava tunni järjekorranumber kuus ja P tarbitud võimsuse keskmine selle tunni jooksul ($0 \leqslant P \leqslant 1000$). Tundide numeratsioon algab iga kuu alguses ühest. Kanded on failis esitatud kronoloogilises järjekorras ja kanded nende tundide kohta, kui klient elektrit ei tarbinud, võivad ka puududa.

Väljund. Tekstifaili `tarb.val` ainsal real on kaks tühikuga eraldatud täisarvu: vastavalt päevase ja öise elektritarbimise summa sisendis kirjeldatud andmete põhjal.

Näide.

`tarb.sis`
```
2012 1
5
1 2
2 3
26 3
32 6
744 0
```

`tarb.val`
```
6 8
```

Sisendis kirjeldatud andmete tähendus on järgmine:

Tund	Võimsus	Kuupäev	Nädalapäev	Kellaaeg	Liik
1	2 kW	1. jaanuar	pühapäev	0:00–1:00	öine
2	3 kW	1. jaanuar	pühapäev	1:00–2:00	öine
26	3 kW	2. jaanuar	esmaspäev	1:00–2:00	öine
32	6 kW	2. jaanuar	esmaspäev	7:00–8:00	päevane
744	0 kW	31. jaanuar	teisipäev	23:00–24:00	öine

Seega on kliendi päevane tarbimine 6 kWh ja öine tarbimine $2 + 3 + 3 = 8$ kWh.

Hindamine. Selles ülesandes tuleb lahendusena esitada üks ZIP-fail, mille sees on kuni 15 sisendfaili nimedega `tarbtest.01.sis`, `tarbtest.02.sis`, ... ja neile vastavad väljundfailid nimedega `tarbtest.01.val`, `tarbtest.02.val`, ...

Kui mõni failipaar ei ole korrekne sisendfail ja sellele vastav korrektne väljundfail, siis selle paari eest punkte ei anta. Korrektseid failipaare kasutatakse teadaolevate vigadega programmide testimiseks ja võistleja saab kindla arvu punkte iga programmi eest, mis annab vähemalt ühes selle võistleja koostatud testis vale vastuse.

7. Kahendpuud

Analüüs 50 punkti

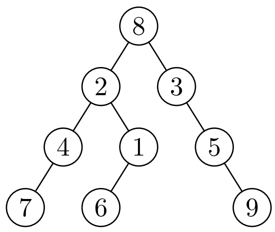
Olgu T kahendpuu[1], mille N tippu on (mingis määramata järjekorras) nummerdatud $1 \ldots N$. Vaatleme skeemi, kus puu tipu t töötlemiseks tehakse järgmised sammud[2]:

1. ...
2. Kui tipul t on vasak alluv v, töödelda v rekursiivselt sama algoritmi järgi.
3. ...
4. Kui tipul t on parem alluv p, töödelda p rekursiivselt sama algoritmi järgi.
5. ...

Nimetame puu T tippude eesjärjestuseks $E(T)$ jada, mille saame, kui väljastame eeltoodud skeemis sammul 1 jooksva tipu t numbri (ja sammudel 3 ja 5 ei tee midagi) ning alustame algoritmi täitmist puu juurtipust.

Analoogiliselt nimetame keskjärjestuseks $K(T)$ jada, mille saame, kui väljastame tipu t numbri sammul 3 ja lõppjärjestuseks $L(T)$ jada, mille saame, kui väljastame tipu t numbri sammul 5.

Näiteks kahendpuu



tippude eesjärjestus on 8, 2, 4, 7, 1, 6, 3, 5, 9, keskjärjestus 7, 4, 2, 6, 1, 8, 3, 5, 9 ja lõppjärjestus 7, 4, 6, 1, 2, 9, 5, 3, 8.

Analüüsida, millised kahendpuu järjestuste komplektid on piisavad puu struktuuri üheseks taastamiseks. See tähendab, vastata küsimusele

Kas komplekti X teadmine võimaldab alati T üheselt taastada?

kõigi võimalike järjestuste komplektide X korral (s.t. $X = \langle E(T) \rangle$, ..., $X = \langle E(T); K(T) \rangle$, ..., $X = \langle E(T); K(T); L(T) \rangle$ korral).

Kui vastus on jaatav, kirjeldada puu taastamise algoritmi. Algoritmid võib anda sõnalise kirjeldusena, kuid kirjeldus peab olema piisavalt täpne algoritmi üheseks mõistmiseks, näiteks:

1. Olgu $E(T) = e_1, e_2, \ldots, e_N$ ja $L(T) = l_1, l_2, \ldots, l_N$.
2. $E(T)$ definitsioonist teame, et e_N ...
3. Selle põhjal saame leida sellise l_i, mis ...

...

Kui vastus on eitav, tuua vähemalt üks kontranäide, see tähendab näidata kaks puud T_1 ja T_2, mille korral kõik vastavad tipujärjestused on samad, kuid T_1 ja T_2 on erinevad.

Hindamine. Selles ülesandes tuleb lahendusena esitada üks tekstifail, milles on põhjendatud vastused kõigi võimalike järjestusekomplektide jaoks.

Hindamisel võetakse punkte maha käsitlemata juhtude, valede vastuste ja puudulike põhjenduste (jaatavate vastuste korral puuduvad või vigased algoritmid, eitavate vastuste korral puuduvad või vigased kontranäited) eest.

[1]Vt. ka `http://www.ut.ee/~at/prog/prog08.pdf`, lõik 8.1.2.
[2]Vt. ka `http://www.ut.ee/~at/prog/prog08.pdf`, lõik 8.2.2.