

1. Tulpdiagramm

1 sekund 20 punkti

Oluline samm graafikute koostamisel on telgedele skaalapunktide valimine. Et saadav graafik oleks ülevaatlik, peab skaalapunkte olema paras arv ja need peavad vastama “ümmargustele” väärtustele.

Selles ülesandes vaatleme tulpdiagrammi Y-telje skaalapunktide valimist. Sisendis on antud tulpade kõrgused ja vaja on valida skaalapunktid nii, et:

- skaala samm (skaala naaberpunktide vahe) s oleks kas $1 \cdot 10^k$, $2 \cdot 10^k$ või $5 \cdot 10^k$, kus k on mingi naturaalarv;
- punktidele vastavad väärtused oleks skaala sammu s täiskordsed;
- madalaim tulp oleks kõrgem kui esimene, kuid mitte kõrgem kui teine skaalapunkt ja kõrgeim tulp oleks madalam kui viimane, kuid mitte madalam kui eelviimane skaalapunkt;
- skaala samm oleks vähim, mille korral punktide arv ei ületa 10.

Kirjutada programm, mis leiab tulpade kõrguste põhjal sobivad skaalapunktide väärtused.

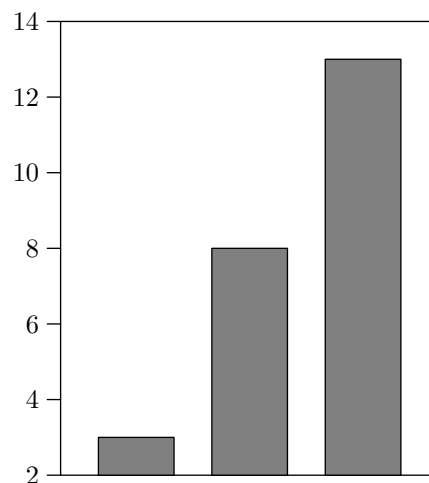
Sisend. Tekstifaili `tulp.sis` esimesel real on tulpade arv N ($1 \leq N \leq 20$) ja teisel real N tühikutega eraldatud täisarvu $0 \dots 10^6$: tulpade kõrgused.

Väljund. Tekstifaili `tulp.val` ainsale reale väljastada maksimaalselt 10 tühikutega eraldatud täisarvu: leitud skaalapunktide väärtused.

Näide.

<code>tulp.sis</code>	<code>tulp.val</code>
3	2 4 6 8 10 12 14
3 8 13	

Valitud skaalapunktidele vastav diagramm:



2. Tabel

1 sekund 30 punkti

Liitmis- ja lahutamistehete harjutamiseks kasutatakse mängu, mille lõpptulemus on arvudega täidetud kolmnurkne tabel, kus alates alt teisest reast on iga arv kahe temast diagonaalis allpool oleva arvu summa:

			20		
		8		12	
	3		5		7
1	2		3		4

Mängu alguses on tabeli mõned elemendid antud ja õpilane peab täitma ülejäänud kohad. Üldjuhul oleks selleks vaja lahendada lineaarvõrrandite süsteem, aga õpilastele antakse harjutamiseks ainult selliseid juhte, kus lahend on kindlasti olemas ja ühene ning kõigi puuduvate arvude leidmiseks piisab sammudest, kus leitakse kahe liidetava järgi summa või summa ja ühe liidetava järgi teine liidetav. Näiteks kui ülaltoodud kolmnurgast on antud arvud 8, 12, 7 ja 1, siis võime 8 ja 12 liitmise teel saada 20 ning edasi lahutamise teel 5, 3, 2, 3 ja 4.

Kirjutada programm, mis lahendab selliseid ülesandeid.

Sisend. Tekstifaili `tabel3.sis` esimesel real on tabeli ridade arv N ($1 \leq N \leq 250$). Järgmisel N real on igaühel kõigepealt tabeli reas teadaolevate elementide arv M_i ($0 \leq M_i \leq i$) ja seejärel M_i arvupaari: antud elemendi järjekorranumber reas ja selle elemendi väärtus. Tabeli i . rea elemendid on nummerdatud vasakult paremale $1 \dots i$.

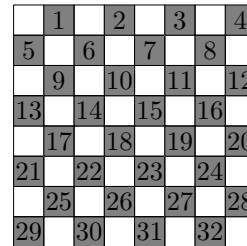
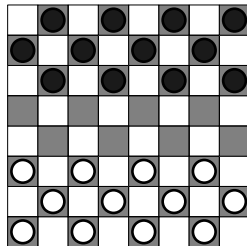
Väljund. Tekstifaili `tabel3.val` väljastada täpselt N rida. Faili i . reale väljastada täpselt i tühikutega eraldatud täisarvu: tabeli i . rea elemendid vasakult paremale. Arve joondada ei ole vaja. Võib eeldada, et kõik arvud mahuvad 32-bitisesse märgiga täisarvu.

Näide.	<code>tabel3.sis</code>	<code>tabel3.val</code>
	4	20
	0	8 12
	2 1 8 2 12	3 5 7
	1 3 7	1 2 3 4
	1 1 1	

3. Kabe

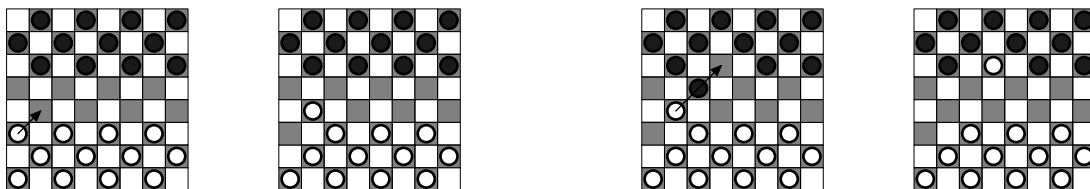
1 sekund 50 punkti

Kabe on kahe mängija lauamäng, mida mängitakse 8×8 laual. Üks mängija mängib mustade, teine valgete nuppudega, mis algseisus on laua kolme esimese ja kolme viimase rea mustadel ruutudel, nagu näha alloleval joonisel vasakul. Kuna kabes käiakse ainult diagonaalselt ja nupud seega kunagi valgetele ruutudele ei satu, nummerdatakse seisude ja käikude märkimiseks ainult musti ruute, nagu on näidatud alloleval joonisel paremal.

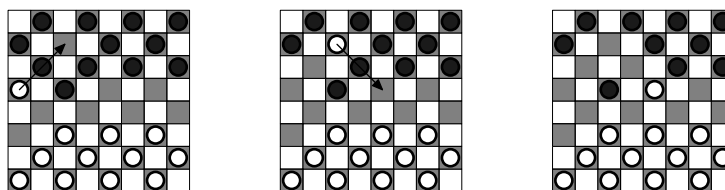


Tavalisel käigul võib mängija nihutada ühe oma nupu ühe ruudu võrra diagonaalis edasi, nagu näidatud alloleval joonisel vasakul. Sellist käiku märgitakse kujul $A-B$, kus A on lähteruudu ja B sihtruudu number; seega joonisel näidatud käik on $21-17$. Käia saab ainult ettepoole (vastase suunas).

Kui mängija nupu kõrval seisab vastase nupp, mille taga on vaba ruut, peab mängija vastase nupust üle hüpates selle lööma, nagu näidatud joonisel paremal. Sellist käiku märgitakse kujul AxB ; seega joonisel näidatud löömine on $17x10$. Lüüa saab nii ette- kui ka tahapoole.



Ühe käiguga võib ühe nupuga järjest hüpates lüüa ka mitu vastase nuppu, nagu näha alloleval joonisel. Sellise käigu korral märgitakse kõik löömisid järjest; seega joonisel näidatud kahe-sammuline löömine on $13x6x15$.



Kirjutada programm, mis loeb laua kirjelduse enne ja pärast käiku ning väljastab tehtud käigu.

Sisend. Tekstifailis `kabe2.sis` on täpselt kaheksa rida, igal real täpselt 17 märki. Kaheksa vasakpoolset veergu kirjeldavad laua seisut enne käiku ja kaheksa parempoolset veergu seisut pärast käiku. Laua seisut kirjelduses tähistab punkt tühja ruutu ning `m` musta ja `v` valget nuppu (kõik nupud on tavalised kivid, kabesid laual ei ole ja käimisel tavalise kivi kabeks muutumise reegleid ei arvestata). Keskmine veerg koosneb tühikutest.

Väljund. Tekstifaili `kabe2.val` esimesele reale väljastada käigu teinud pool (sõna **MUST** või **VALGE**) ja teisele reale tehtud käik eelpool kirjeldatud kujul. Võib eeldada, et tehti üks käik. Kui võimalikke vastuseid on mitu, väljastada neist ükskõik milline.

Näide.	<code>kabe2.sis</code>	<code>kabe2.val</code>
	<code>.m.m.m.m .m.m.m.m</code>	<code>VALGE</code>
	<code>m.m.m.m. m.m.m.m.</code>	<code>21-17</code>
	<code>.m.m.m.m .m.m.m.m</code>	
	<code>.....</code>	
	<code>..... .v.....</code>	
	<code>v.v.v.v. .v.v.v.v.</code>	
	<code>.v.v.v.v .v.v.v.v</code>	
	<code>v.v.v.v. v.v.v.v.</code>	

1. Matk

1 sekund 20 punkti

Kaks sõpra tahavad koos K päevaks matkama minna. Kuna nad on mõlemad aktiivsed inimesed, on neil juba üsna paljudeks päevadeks muud plaanid tehtud.

Kirjutada programm, mis leiab sõprade kalendrite põhjal varaseima võimaluse matkale minna.

Sisend. Tekstifaili `matk.sis` esimesel real on esimesel sõbral kinni olevate päevade arv N_1 ($0 \leq N_1 \leq 300$) ja järgmisel N_1 real igaühel ühe sellise päeva kuupäev kujul $KK PP$; järgmisel real on teisel sõbral kinni olevate päevade arv N_2 ($0 \leq N_2 \leq 300$) ja järgmisel N_2 real igaühel ühe sellise päeva kuupäev kujul $KK PP$. Võib eeldada, et kõik kuupäevad on korrektsed 2012. aasta kuupäevad. Faili viimasel real on täisarv K ($1 \leq K \leq 300$) — matka pikkus päevades.

Väljund. Tekstifaili `matk.val` ainsale reale väljastada kõige varasem kuupäev, millal sõbrad võivad matkale asuda (varasem selline kuupäev, et sellest alates on mõlemal K päeva järjest vabad), või tekst `EI SAA`, kui sõpradel pole terve 2012. aasta jooksul K järjestikust ühist vaba päeva. Kuupäev väljastada kujul $KK PP$ (kahekohaline kuu number, tühik, kahekohaline päeva number).

Näide.	<code>matk.sis</code>	<code>matk.val</code>
	2	01 04
	01 01	
	01 02	
	2	
	01 03	
	01 01	
	3	

Hindamine. Selles ülesandes antakse `EI SAA` vastusega testide eest punkte ainult neile programmidele, mis lahendavad õigesti vähemalt ühe testi, kus sõbrad saavad koos matkale minna.

2. ETSO koodid

1 sekund 30 punkti

Euroopa elektrivõrkudes on kõigile objektidele omistatud unikaalsed 16-kohalised koodid. Koodi viimane märk on kontrollsümbol, mis võimaldab tuvastada andmete sisestamisel tehtud vigu. Kontrollsümboli arvutamise algoritmi kirjeldus on toodud lisas.

Kirjutada programm, mis tuvastab, kas sisendis antud koodid on korrektsed.

Sisend. Tekstifaili `etso.sis` esimesel real on kontrollitavate koodide arv N ($1 \leq N \leq 100$) ja järgmisel N real igaihel täpselt 16 märki hulgast $\{0 \dots 9, A \dots Z, -\}$ — üks uuritav kood.

Väljund. Tekstifaili `etso.val` väljastada täpselt N rida, igale reale sõna JAH või EI vastavalt sellele, kas sisendis antud kood on korrektne või mitte. Vastused väljastada koodide sisendis esinemise järjekorras.

Näide.	<code>etso.sis</code>	<code>etso.val</code>
	3	JAH
	11XRWENET12345-2	JAH
	11Y123456789012T	EI
	10Z317973010277Q	

Lisa. Alljärgnev algoritmi kirjeldus on tõlgitud dokumendist “A Common Identification System For The Energy Industry: The Energy Identification Coding Scheme EIC (Version 4, Revision 3, 08 June 2009)”, © ETSO 2002–2008:

Sissejuhatus

Käesolev dokument kirjeldab ETSO koodide korrektsuse kontrollimise algoritmi. Iga ETSO kood on varustatud kontrollsümboliga. Kontrollsümbol lisatakse koodi lõppu ja selle abil on võimalik kontrollida koodi korrektsust. Kontrollsümbol arvutatakse koodi ülejäänud märkidele lihtsa algoritmi rakendamise teel. Selle algoritmi täitmisel saadud kontrollsümbolit koodis leiduvaga võrreldes on võimalik tuvastada, et sisestatud kood on täielik ja korrektne. Selle info võimalikud kasustusjuhud:

- kasutaja sisestatud (või skaneeritud) koodide korrektsuse kontroll enne nende kasutamist;
- uute koodide väljastamine.

ETSO kood

ETSO koodid on fikseeritud pikkusega tärgkoodid. Kood sisaldab infot selle välja andnud asutuse ja identifitseeritava objekti liigi kohta. Osapoolte ja piirkondade koodid on 16-kohalised tärgkoodid. Koodi viimane sümbol on kontrollsümbol, mis arvutatakse koodi ülejäänud märkidest ETSO algoritmiga. Osapoolte kood võib olla näiteks 11XRWENET12345-2 ja piirkona kood 11Y1234567890123.

Nende koodide viimased märgid (vastavalt 2 ja 3) ongi käesolevaga kirjeldatavad kontrollsümbolid.

Kontrollsümboli arvutamine. Üldine algoritm kõigile koodidele

Samm 1: Eraldatakse koodi 15 esimest märki, näiteks:

1	1	X	R	W	E	N	E	T	1	2	3	4	5	-
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Samm 2: Kui koodis esineb tähti, asendatakse need nende arvuliste väärtustega vastavalt tabelile:

Märk	0	1	2	3	4	5	6	7	8	9
Väärtus	0	1	2	3	4	5	6	7	8	9

Märk	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Väärtus	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Märk	O	P	Q	R	S	T	U	V	W	X	Y	Z	-
Väärtus	24	25	26	27	28	29	30	31	32	33	34	35	36

Näiteks:

1	1	X	R	W	E	N	E	T	1	2	3	4	5	-
1	1	33	27	32	14	23	14	29	1	2	3	4	5	36

Samm 3: Seejärel antakse koodi igale positsioonile kaal, alustades suurima väärtusega vasakult ja lõpetades ühega paremal:

1	1	33	27	32	14	23	14	29	1	2	3	4	5	36
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Samm 4: Iga positsiooni väärtus korrutatakse selle kaaluga:

1	1	33	27	32	14	23	14	29	1	2	3	4	5	36
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
16	15	462	351	384	154	230	126	232	7	12	15	16	15	72

Samm 5: Korrutised summeeritakse:

16	15	462	351	384	154	230	126	232	7	12	15	16	15	72
----	----	-----	-----	-----	-----	-----	-----	-----	---	----	----	----	----	----

Summa: 2107.

Samm 6: Saadud summa 2107 taandatakse mooduli 37 järgi (mis vastab kasutada olevate märkide koguarvule) valemiga $(36 - \text{MOD}((2107 - 1), 37))$, kus $\text{MOD}(X, Y)$ on arvu X arvuga Y jagamisel tekkiv jääk.

Avaldise väärtus on 2. Kuna see on väiksem kui 10, ongi see kontrollsümbol. Kui väärtus oleks olnud suurem kui 9, oleks see teisendatud tagasi sümboliks sammus 2 kirjeldatud vastavustabeli abil.

Täielik kood on seega 11XRWENET12345-2.

Kui saadud kontrollsümbol on “-” (avaldise väärtus 36), siis tuleb koodis mõni märk teisega asendada, et saada kontrollsümboli avaldis, mille väärtus ei ole 36.

Tugevused

Nagu kõik järjestikuste kaaludega süsteemid, tuvastab see meetod kõik ühe märgi valesti sisestamise ja kõik kahe kõrvutioleva märgi vahetamise vead. Seega tuvastaks see meetod, et kood 10Z317973010277Q on vigane.

See algoritm on väga kasulik selle poolest, et lubab koodides lisaks numbritele kasutada ka tähti ja seega avardab märkimisväärselt kasutada olevate koodide koguarvu.

3. Tabel

1 sekund 50 punkti

Vaatleme kolmnurkset arvutabelit, kus alates alt teisest reast on iga arv kahe temast diagonaalis allpool oleva arvu summa:

			20		
		8		12	
	3		5		7
1	2		3		4

Kirjutada programm, mis saab tabeli iga rea ühe elemendi väärtuse ja taastab selle info põhjal terve tabeli.

Sisend. Tekstifaili `tabel2.sis` esimesel real on tabeli ridade arv N ($1 \leq N \leq 100$). Järgmisel N real on igaühel kaks täisarvu: antud elemendi järjekorranumber reas ja selle elemendi väärtus. Tabeli i . rea elemendid on nummerdatud vasakult paremale $1 \dots i$.

Väljund. Tekstifaili `tabel2.val` väljastada täpselt N rida. Faili i . reale väljastada täpselt i tühikutega eraldatud täisarvu: tabeli i . rea elemendid vasakult paremale. Arve joondada ei ole vaja. Võib eeldada, et kõik arvud mahuvad 32-bitisesse märgiga täisarvu.

Näide.	<code>tabel2.sis</code>	<code>tabel2.val</code>
	4	20
	1 20	8 12
	1 8	3 5 7
	1 3	1 2 3 4
	1 1	

1. Kino

1 sekund 20 punkti

Kaks sõpra tahavad koos kinno minna. Kuna nad on mõlemad aktiivsed inimesed, on neil juba üsna paljudeks õhtuteks muud plaanid tehtud.

Kirjutada programm, mis leiab sõprade kalendrite põhjal esimese mõlemal vaba õhtu.

Sisend. Tekstifaili `kino.sis` esimesel real on esimesel sõbral kinni olevate õhtute arv N_1 ($0 \leq N_1 \leq 300$) ja järgmisel N_1 real igaühel ühe sellise õhtu kuupäev kujul `KK PP`; järgmisel real on teisel sõbral kinni olevate õhtute arv N_2 ($0 \leq N_2 \leq 300$) ja järgmisel N_2 real igaühel ühe sellise õhtu kuupäev kujul `KK PP`. Võib eeldada, et kõik kuupäevad on korrektsed 2012. aasta kuupäevad.

Väljund. Tekstifaili `kino.val` ainsale reale väljastada esimese mõlemal sõbral vaba õhtu kuupäev kujul `KK PP` (kahekohaline kuu number, tühik, kahekohaline päeva number), või tekst `EI SAA`, kui sõpradel pole terve 2012. aasta jooksul ühtki ühist vaba õhtut.

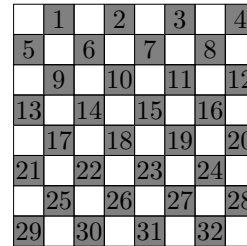
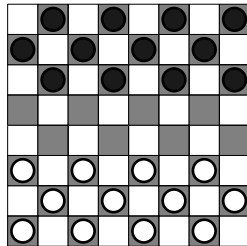
Näide.	<code>kino.sis</code>	<code>kino.val</code>
	2	01 04
	01 01	
	01 02	
	2	
	01 03	
	01 01	

Hindamine. Selles ülesandes antakse `EI SAA` vastusega testide eest punkte ainult neile programmidele, mis lahendavad õigesti vähemalt ühe testi, kus sõbrad saavad koos kinno minna.

2. Kabe

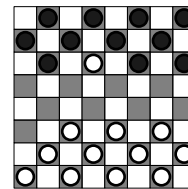
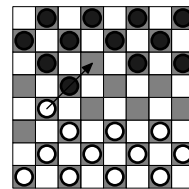
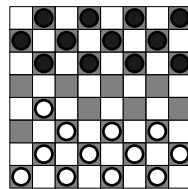
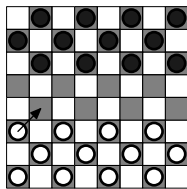
1 sekund 30 punkti

Kabe on kahe mängija lauamäng, mida mängitakse 8×8 laual. Üks mängija mängib mustade, teine valgete nuppudega, mis algseisus on laua kolme esimese ja kolme viimase rea mustadel ruutudel, nagu näha alloleval joonisel vasakul. Kuna kabes käiakse ainult diagonaalselt ja nupud seega kunagi valgetele ruutudele ei satu, nummerdatakse seisude ja käikude märkimiseks ainult musti ruute, nagu on näidatud alloleval joonisel paremal.



Tavalisel käigul võib mängija nihutada ühe oma nupu ühe ruudu võrra diagonaalis edasi, nagu näidatud alloleval joonisel vasakul. Sellist käiku märgitakse kujul $A-B$, kus A on lähteruudu ja B sihtruudu number; seega joonisel näidatud käik on $21-17$. Käia saab ainult ettepoole (vastase suunas).

Kui mängija nupu kõrval seisab vastase nupp, mille taga on vaba ruut, peab mängija vastase nupust üle hüpates selle lööma, nagu näidatud joonisel paremal. Sellist käiku märgitakse kujul AxB ; seega joonisel näidatud löömine on $17x10$. Lüüa saab nii ette- kui ka tahapoole.



Kirjutada programm, mis loeb laua kirjelduse enne ja pärast käiku ning väljastab tehtud käigu.

Sisend. Tekstifailis `kabe1.sis` on täpselt kaheksa rida, igal real täpselt 17 märki. Kaheksa vasakpoolset veergu kirjeldavad laua seisut enne käiku ja kaheksa parempoolset veergu seisut pärast käiku. Laua seisut kirjelduses tähistab punkt tühja ruutu ning `m` musta ja `v` valget nuppu (kõik nupud on tavalised kivid, kabesid laual ei ole). Keskmise veergu koosneb tühikutest.

Väljund. Tekstifaili `kabe1.val` esimesele reale väljastada käigu teinud pool (sõna `MUST` või `VALGE`) ja teisele reale tehtud käik eelpool kirjeldatud kujul. Võib eeldada, et tehti üks tavaline käik või löödi üks vastase nupp.

Näide.

```
kabe1.sis
.m.m.m.m .m.m.m.m
m.m.m.m. m.m.m.m.
.m.m.m.m .m.m.m.m
.....
..... .v.....
v.v.v.v. .v.v.v.v.
.v.v.v.v .v.v.v.v
v.v.v.v. v.v.v.v.
```

```
kabe1.val
VALGE
21-17
```

3. Tabel

1 sekund

50 punkti

Vaatleme kolmnurkset arvutabelit, kus alates alt teisest reast on iga arv kahe temast diagonaalis allpool oleva arvu summa:

```

      20
     8  12
    3  5  7
   1  2  3  4
```

Kirjutada programm, mis saab tabeli iga rea vasakpoolseima elemendi väärtuse ja taastab selle info põhjal terve tabeli.

Sisend. Tekstifaili `tabel1.sis` esimesel real on tabeli ridade arv N ($1 \leq N \leq 100$). Järgmisel N real on igaühel üks täisarv: tabeli ridade vasakpoolseimad elemendid.

Väljund. Tekstifaili `tabel1.val` väljastada täpselt N rida. Faili i . reale väljastada täpselt i tühikutega eraldatud täisarvu: tabeli i . rea elemendid vasakult paremale. Arve joondada ei ole vaja. Võib eeldada, et kõik arvud mahuvad 32-bitisesse märgiga täisarvu.

Näide.	tabel1.sis	tabel1.val
	4	20
	20	8 12
	8	3 5 7
	3	1 2 3 4
	1	