

1. Doominokivid

1 sekund 30 punkti

N -doomino kivi koosneb kahest ruudukujulisest poolest, millel on kummalgi $0 \dots N$ silma. Doomino-komplektis on kõik võimalikud erinevad kivid, kusjuures kivi, mille ühel poolel on a ja teisel poolel b silma, on komplektis ühes eksemplaris (s.t kive $a : b$ ja $b : a$ ei loeta erinevateks).

On antud mingi hulk N -doomino kive. Kirjutada programm, mis leiab kivid, mis tuleks antud kividele lisada, et saada täiskomplekt.

Sisend. Tekstifaili `dom.sis` esimesel real on täisarv N ($0 \leq N \leq 9$) ja antud kivide arv K ($0 \leq K \leq (N+1)(N+2)/2$). Järgmisel K real on antud kivide kirjeldused kujul $A_i B_i$ ($0 \leq A_i, B_i \leq N$). Kivid on antud suvalises järjekorras, aga on teada, et ükski kivi ei esine sisendis korduvalt.

Väljund. Tekstifaili `dom.val` esimesele reale väljastada puuduvate kivide arv P ja järgmisele P reale nende kivide kirjeldused kujul $A_i B_i$. Kivide järjekord failis, samuti A_i ja B_i järjekord real pole olulised.

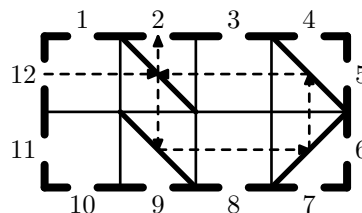
Näide.	dom.sis	dom.val
	3 4	6
	0 0	0 1
	1 1	0 2
	2 2	0 3
	3 3	1 2
		1 3
		2 3

3-doomino komplektis on kokku 10 kivi: 0:0, 0:1, 0:2, 0:3, 1:1, 1:2, 1:3, 2:2, 2:3, 3:3.

2. Valguskiir

1 sekund 40 punkti

Vaatleme joonisel kujutatud “maagilist” karpi, mille põhi on jagatud ruudukujulisteks väljadeks. Igas ruudus võib olla maksimaalselt üks kahepoolne peegel. Iga peegel ulatab oma ruudu ühest nurgast selle vastasnurka (ja on seega karbi külgede suhtes 45° nurga all). Karbi iga rea ja iga veeru mõlemas otsas on auk. Augud on nummerdatud $1 \dots 2 \cdot (R + V)$, nagu näha joonisel.



Ühest august lastakse karbi küljega täisnurga all sisse valguskiir, mis võib peeglitest peegelduda ja mõne aja pärast mingi augu kaudu karbist väljub. Kirjutada programm, mis leiab, millise augu kaudu kiir karbist väljub.

Sisend. Tekstifaili `kiir.sis` esimesel real on peeglikarbi ridade arv R ($1 \leq R \leq 50$) ja veergude arv V ($1 \leq V \leq 50$). Järgmisel R real on igaühel täpselt V märki: karbi kirjeldus, kus # tähistab tühja ruutu ning \ ja / peegleid. Faili viimasel real on valguskiire sisenemisaugu number S .

Väljund. Tekstifaili `kiir.val` esimesele reale väljastada kiire karbis peegeldumiste arv N ja teisele reale tema karbist väljumise augu number T .

Näide.	kiir.sis	kiir.val
	2 4	5
	#\#\	2
	#\#/	
	12	

Hindamine. N ja T väljastamist hinnatakse eraldi, kumbki annab 50% testi väärtusest. Kui programm ühte neist kahest arvust ei leia, väljastada selle asemel -1 .

3. Pank

1 sekund

30 punkti

Pangakontoris on K kassat (tähistatud $1 \dots K$), mis kõik kokku pakuvad klientidele T teenust (tähistatud $1 \dots T$). Iga kassa kohta on teada, milliseid teenuseid see kassa pakub, kusjuures iga teenust võib pakkuda üks või mitu kassat.

Pangakontoris on järjekorraautomaat, millel on T nuppu: üks nupp iga teenuse jaoks. Panka tulnud klient vajutab soovitava teenuse nuppu ja automaat paneb ta järjekorra lõppu. Kui mingi kassa vabaneb, suunab järjekorraautomaat sellesse kassasse kõige varem järjekorda asunud kliendi, kes soovib mõnda selles kassas pakutavat teenust. Kui kliendile sobivad mitu kassat, suunab järjekorraautomaat ta alati vähima numbriga kassasse.

Kirjutada programm, mis saab andmed klientide pankas saabumise järjekorra ning igaühe soovitava teenuse ja tema teenindamiseks kuluva aja kohta ning väljastab, milline kassa ja mis ajal iga klienti teenindas.

Sisend. Tekstifaili `pank.sis` esimesel real on kassade arv K ($1 \leq K \leq 25$) ja teenuste koguarv T ($1 \leq T \leq 25$). Järgmised K rida kirjeldavad kassade pakutavaid teenuseid. Real number $i+1$ on kõigepealt kassa number i pakutavate teenuste arv T_i ($1 \leq T_i \leq T$) ja selle järel T_i täisarvu: nende teenuste tähised. Järgmisel real (number $K+2$) on pangakontorit päeva jooksul külastavate klientide arv N ($1 \leq N \leq 10\,000$). Järgmised N rida kirjeldavad kliente nende pankas saabumise järjekorras. Real number $K+i+2$ on kliendi number i pankas saabumise aeg A_i ($0 \leq A_i \leq 10\,000$), tema soovitava teenuse tähis B_i ($1 \leq B_i \leq T$) ja tema teenindamiseks kassas kuluv aeg C_i ($1 \leq C_i \leq 10\,000$). On teada, et mitte mingid kaks klienti ei saabu pankas täpselt samal ajal ($A_i < A_{i+1}$ iga $i < N$ korral).

Väljund. Tekstifaili `pank.val` väljastada täpselt N rida. Reale number i väljastada klienti number i teenindanud kassa tähis ja kliendi kassasse pääsemise kellaaeg. Kassiriid puhkepause ei tee ja järjekorras ootav klient jõuab kassani kohe kui see vabaneb (kui klient, kelle teenindamise ajakulu on c , läheb kassasse ajahetkel d , siis järgmine klient saab sinna kassasse ajahetkel $d+c$).

Näide.	<code>pank.sis</code>	<code>pank.val</code>
	2 3	1 0
	2 1 2	2 5
	2 2 3	1 10
	4	1 20
	0 1 10	
	5 2 10	
	7 2 10	
	9 1 10	

Klient 1 saabub pankas ajahetkel 0, soovib teenust 1 ja läheb kohe kassasse 1. Klient 2 saabub pankas ajahetkel 5, soovib teenust 2 ja läheb kohe kassasse 2. Klient 3 saabub pankas ajahetkel 7 ja soovib teenust 2, aga mõlemad seda teenust pakuvad kassad on hõivatud ja klient jääb ootama. Klient 4 saabub pankas ajahetkel 9 ja soovib teenust 1, aga seda pakuv kassa 1 on hõivatud ja klient jääb ootama.

Ajahetkel 10 lõpetab kassa 1 kliendi 1 teenindamise ja klient 3 läheb sellesse kassasse. Ajahetkel 15 lõpetab kassa 2 kliendi 2 teenindamise, aga kuna see kassa ei paku teenust 1, mida soovib ainsana järjekorras olev klient 4, jääb see kassa lihtsalt tühjaks. Ajahetkel 20 lõpetab kassa 1 kliendi 3 teenindamise ja klient 4 läheb sellesse kassasse.

1. Prefikspakkimine

1 sekund 30 punkti

Prefikspakkimine on kahesammuline algoritm sõnede kogumi pakkimiseks:

- Sorteerime sõned tähestikuliselt.
- Tulemuse igas järjestikuste sõnede paaris $a_1a_2 \dots a_{n_a}, b_1b_2 \dots b_{n_b}$ leiame suurima sellise k , et iga $1 \leq i \leq k$ korral $a_i = b_i$. Kui $k > 0$, avaldame teise sõne kujul $kb_{k+1}b_{k+2} \dots b_{n_b}$.
Teisisõnu: igas sõnes väljastame talle eelnevaga ühise mittetühja prefiksi asemel selle pikkuse (kümnnendsüsteemis). Lahtipakkimisel saame sõne kergesti taastada, kuna meil on teada nii eelnev sõne kui ka sellest üle võetava prefiksi pikkus.

Kirjutada programm, mis leiab antud sõnekogumi prefikspakitud kuju.

Sisend. Tekstifaili `pref.sis` esimesel real on sõnede arv N ($1 \leq N \leq 10\,000$) ja järgmisel N real igalühel üks väikestest ladina tähtedest koosnev sõne pikkusega $1 \dots 30$ märki.

Väljund. Tekstifaili `pref.val` väljastada täpselt N rida: pakitud sõnekogum, iga sõne eraldi reale. Sõnede sorteerimisel võrrelda märke nende ASCII koodide alusel.

Näide.	<code>pref.sis</code>	<code>pref.val</code>
	4	kala
	post	2ss
	kass	3t
	kala	post
	kast	

Sõnede `post`, `kass`, `kala`, `kast` tähestikuline järjekord on `kala`, `kass`, `kast`, `post`, millest näeme, et sõnedel `kala` ja `kass` on ühine prefiks `ka` pikkusega 2, sõnedel `kass` ja `kast` ühine prefiks `kas` pikkusega 3 ning sõnedel `kast` ja `post` ühist prefiksit ei olegi. Leitud prefiksitate asendamisel nende pikkustega saamegi tulemuseks `kala`, `2ss`, `3t`, `post`.

2. Palindroomarvud

1 sekund 30 punkti

Täisarvu nimetatakse palindroomiks, kui ta on sama nii vasakult paremale kui ka paremalt vasakule lugedes. Näiteks 121 on palindroom, aga 123 ei ole.

Kirjutada programm, mis leiab antud arvule lähimad palindroomid.

Sisend. Tekstifaili `pal.sis` ainsal real on täisarv N ($0 \leq N \leq 9\,223\,372\,036\,302\,733\,229$).

Väljund. Tekstifaili `pal.val` esimesele reale väljastada täisarv N_1 : maksimaalne palindroom, mis ei ole suurem kui N , ja teisele reale täisarv N_2 : minimaalne palindroom, mis ei ole väiksem kui N .

Näide.	<code>pal.sis</code>	<code>pal.val</code>
	123	121
		131

Hindamine. N_1 ja N_2 väljastamist hinnatakse eraldi, kumbki annab 50% testi väärtusest. Kui programm ühte kahest palindroomist ei leia, väljastada selle asemel arv -1 .

3. Tsirkus

1 sekund 40 punkti

Lauamängu “Tsirkus” laud koosneb N ruudust, mis on nummerdatud $1 \dots N$. Ruut number 1 on algus- ja ruut number N lõpuruut. Mängu alguses asetatakse mängija nupp algusruudule ja tema eesmärk on jõuda lõpuruudule.

Igal käigul viskab mängija tavalist 6-tahulist täringut ja käib oma nupuga saadud silmade arvu ($1 \dots 6$) võrra edasi (s silma viskamise järel liigub nupp ruudult r ruudule $r + s$).

Osa ruute on omavahel ühendatud ühesuunaliste tunnelitega. Kui mängija nupp peatub täringuviskega määratud liikumise järel mingi tunneli sissepääsul, liigub see kohe edasi tunneli väljapääsule. Kui nupp peatub tunneli väljapääsul, ei juhtu midagi.

Kirjutada programm, mis leiab vähima võimaliku täringuvisete arvu, millega võib jõuda algusruudult lõpuruudule, kui kõik täringuvisked annavad soodsaimad tulemused.

Sisend. Tekstifaili `ts.sis` esimesel real on laua ruutude arv N ($1 \leq N \leq 1000$) ja tunnelite arv M ($0 \leq 2 \cdot M \leq N$). Järgmisel M real on igaühel kaks erinevat täisarvu: ühe tunneli sisse- ja väljapääsu number. On teada, et ükski ruut pole rohkem kui ühe tunneli otspunkt ning algus- ja lõpuruut pole ühegi tunneli sissepääs.

Väljund. Tekstifaili `ts.val` esimesele reale väljastada otsitav täringuvisete arv K ja järgmisele K reale igaühel üks täisarv: vajalikud silmade arvud visete järjekorras. Viimase käigu järel peab nupp peatuma täpselt lõpuruudul, sellest üle minna ei tohi. On teada, et lõpuruudule pääsemine on võimalik. Kui minimaalse visete arvuga lahendusi on mitu, väljastada ükskõik milline neist.

Näide.	<code>ts.sis</code>	<code>ts.val</code>
	16 2	2
	2 10	1
	12 6	6

Kahe täringuviskega saab mängu lõpetada, kui

1. visata täringuga 1 silm, liikuda sellega ruudult 1 ruudule 2 ja sõita sealt tunneliga edasi ruudule 10;
2. visata täringuga 6 silma ja liikuda sellega ruudult 10 ruudule 16.

Peaks olema ilmne, et ühe viskega lõpuruudule jõudmine on võimatu.