

1. Logianalüüs

1 sekund 30 punkti

Arvutisüsteemi logis on kanded kujul “*aeg sündmus sessiooni-ID kasutaja-ID*”, kus *aeg* on sündmuse toimumise aeg ja *sündmus* on kas S (kasutaja logis sisse), V (kasutaja logis välja) või R (kasutaja algatas süsteemi alglaadimise, mille peale logiti kõik kasutajad korraga välja).

Sessioonide identifikaatorid võivad logis korduda, aga kunagi ei ole korraga aktiivsed mitu sama identifikaatoriga sessiooni. Ühel kasutajal võib olla mitu paralleelset sessiooni, aga ka siis on igal sessioonil unikaalne identifikaator.

Kirjutada programm, mis leiab selle logi põhjal iga kasutaja arvutikasutusa ja kogupikkuse logi algusest kuni logi viimase kande ajani, võttes mitme paralleelse sessiooni aega arvesse ühekordselt. Võib eeldada, et logi algab hetkel, kui ükski kasutaja ei ole sisse logitud.

Sisend. Tekstifaili `loge.sis` esimesel real on logi kannete arv N ($1 \leq N \leq 10\,000$) ja järgmisel N real igaühel üks eelkirjeldatud kujul kanne. Nii ajad kui sessioonide ja kasutajate identifikaatorid on täisarvud $0 \dots 10^9$ ja kanded on antud kronoloogilises järjekorras.

Väljund. Tekstifaili `loge.val` väljastada täpselt üks rida iga sisendis esinenud kasutaja kohta. Igale reale väljastada kaks tühikuga eraldatud täisarvu: kasutaja identifikaator ja selle kasutaja arvutikasutuse koguaeg. Ridade järjekord failis pole oluline.

Näide.	<code>loge.sis</code>	<code>loge.val</code>
	3	1 30
	10 S 1 1	2 20
	20 S 2 2	
	40 V 1 1	

Näide.	<code>loge.sis</code>	<code>loge.val</code>
	4	1 30
	10 S 1 1	
	20 S 2 1	
	30 V 2 1	
	40 V 1 1	

2. Initsiaalid

1 sekund 30 punkti

Sõpruskonnal on mõte osaleda suusamaratonil. Treenimise motivatsiooni tõstmiseks tahavad nad joonistada tahvlile suure tabeli, kuhu kõik saavad oma trennikilomeetrid kirja panna. Tabelis on iga inimese kohta üks veerg. Kuna tahvlil on ruumi vähe, aga sõpru on palju, siis tuleb nimesid kuidagi lühendada, et kõik tahvlile ära mahuksid. Mõte on kasutada igaühe ees- ja perenime esitähte ja perenimest lisaks veel nii vähe kui võimalik, et tulemus oleks unikaalne.

Kirjutada programm, mis loeb sõprade nimed ja väljastab sobivad tabeli veergude pealkirjad.

Sisend. Tekstifaili `init.sis` esimesel real on sõprade arv N ($1 \leq N \leq 10\,000$) ja järgmisel N real igaühel ühe sõbra ees- ja perenimi. Võib eeldada, et kõik nimed koosnevad ainult ladina tähtedest ja ühegi ees- ega perenime pikkus ei ületa 30 märki. Samuti võib eeldada, et eesnime esitäht ja terve perenimi kokku on igal sõbral unikaalne.

Väljund. Tekstifaili `init.val` väljastada täpselt N rida. Igale reale väljastada ühe nime lühend nime sisendfailis esinemise järjekorras. Lühendite unikaalsuse kontrollimisel suur- ja väiketähti ei eristata.

Näide.	<code>init.sis</code>	<code>init.val</code>
	3	MMa
	Mari Maasikas	MMu
	Mati Murakas	MK
	Mart Kivikas	

3. Sidefirmade ühinemine

1 sekund 40 punkti

Ühes riigis on mitu mobiilsidefirmat. Igal firmal on hulk kaabelliine, mis ühendavad kõik selle firma tugijaamad üheks võrguks. Nüüd on need firmad otsustanud liituda ja tahavad ka oma võrgud omavahel ühendada. Selleks on nad saanud kaablifirmadelt hulga pakkumisi erinevate tugijaamade vahele uute kaabelliinide paigaldamiseks.

Kirjutada programm, mis aitab mobiilifirmadel leida minimaalse hinnaga pakkumiste komplekti, mille abil oma võrkude võrk üheks tervikuks liita.

Sisend. Tekstifaili `side.sis` esimesel real on kõigi firmade tugijaamade koguarv N ($1 \leq N \leq 10\,000$), olemasolevate kaabelliinide arv M ($0 \leq M \leq 10\,000$) ja uute liinide pakkumiste arv K ($0 \leq K \leq 10\,000$). Tugijaamad on nummerdatud $1 \dots N$.

Järgmisel M real on igaühel kaks täisarvu A_i ja B_i ($1 \leq A_i \leq N$, $1 \leq B_i \leq N$, $A_i \neq B_i$), mis näitavad, et tugijaamade A_i ja B_i vahel juba on kaabelliin. Järgmisel K real on igaühel kolm täisarvu A_j , B_j ja C_j ($1 \leq A_j \leq N$, $1 \leq B_j \leq N$, $A_j \neq B_j$, $0 < C_j \leq 10\,000$), mis näitavad, et on tugijaamade A_j ja B_j vahele saaks ehitada uue kaabelliini hinnaga C_j . Pakkumised on nummerdatud $1 \dots K$ nende sisendis esinemise järjekorras.

Väljund. Tekstifaili `side.val` esimesele reale väljastada vajalike tööde minimaalne koguhind H ja ehitatavate uute liinide arv L . Faili teisele reale väljastada L tühikutega eraldatud täisarvu: nende pakkumiste numbrid, mis tuleks vastu võtta. Pakkumiste järjekord real pole oluline. Kui minimaalse hinnaga lahendusi on mitu, väljastada ükskõik milline neist. Võib eeldada, et võrkude ühendamine on igas testis võimalik.

Näide.	<code>side.sis</code>	<code>side.val</code>
	5 3 2	5 1
	1 2	1
	2 3	
	4 5	
	1 4 5	
	2 5 7	

1. Logianalüüs

1 sekund 30 punkti

Arvutisüsteemi logis on kanded kujul “*aeg sündmus sessiooni-ID kasutaja-ID*”, kus *aeg* on sündmuse toimumise aeg ja *sündmus* on kas **S** (kasutaja logis sisse), **V** (kasutaja logis välja) või **R** (kasutaja algatas süsteemi alglaadimise, mille peale logiti kõik kasutajad korraga välja).

Sessioonide identifikaatorid võivad logis korduda, aga kunagi ei ole korraga aktiivsed mitu sama identifikaatoriga sessiooni. Ühel kasutajal võib olla mitu paralleelset sessiooni, aga ka siis on igal sessioonil unikaalne identifikaator.

Kirjutada programm, mis leiab selle logi põhjal iga kasutaja kõigi sessioonide pikkuste summa logi algusest kuni logi viimase kande ajani ja väljastab need summad kõigi kasutajate jaoks. Võib eeldada, et logi algab hetkel, kui ükski kasutaja ei ole sisse logitud.

Sisend. Tekstifaili `logg.sis` esimesel real on logi kannete arv N ($1 \leq N \leq 10\,000$) ja järgmisel N real igaühel üks eelkirjeldatud kujul kanne. Nii ajad kui sessioonide ja kasutajate identifikaatorid on täisarvud $0 \dots 10^9$ ja kanded on antud kronoloogilises järjekorras.

Väljund. Tekstifaili `logg.val` väljastada täpselt üks rida iga sisendis esinenud kasutaja kohta. Igale reale väljastada kaks tühikuga eraldatud täisarvu: kasutaja identifikaator ja selle kasutaja kõigi sessioonide pikkuste summa. Ridade järjekord failis pole oluline.

Näide.	<code>logg.sis</code>	<code>logg.val</code>
	4	1 30
	10 S 1 1	2 10
	20 S 2 2	
	30 V 2 2	
	40 V 1 1	

Näide.	<code>logg.sis</code>	<code>logg.val</code>
	3	1 30
	10 S 1 1	2 20
	20 S 2 2	
	40 V 1 1	

2. DNS-kodeering

1 sekund 30 punkti

Interneti domeeninimede süsteem DNS on tõstutundetu, näiteks `lists.ut.ee` ja `LiStS.Ut.Ee` on nimedena samaväärsed. Seega on võimalik nimedes sobivalt suur- ja väiketähti kasutades neisse lisainfot kodeerida.

Selles ülesandes on vaja kirjutada kaks programmi, millest üks kodeerib antud nimesse antud lisainfo ja teine dekodeerib esimeselt saadud nimest sinna peidetud lisainfo.

Kodeerija

Sisend. Tekstifaili `dnsx.sis` esimesel real on sisendite arv N ($1 \leq N \leq 1000$). Järgmisel $2 \cdot N$ real on N sisendit, igaüks kahel real: esimesel real DNS-nimi S_i ja teisel real mittenegatiivne täisarv X_i . Nimi on väikeste ladina tähtede, numbrite, miinuste ja punktide jada pikkusega kuni 63 märki. Võib eeldada, et iga X_i on selline, et optimaalse kodeerimisviisi puhul on nimesse S_i võimalik kodeerida vähemalt kõik täisarvud $0 \dots X_i$.

Väljund. Tekstifaili `dnsx.val` väljastada täpselt N rida, igale reale üks nimi S_i , milles mõned väikesed tähed võivad olla asendatud suurtega. Muud muutused nimes ei ole lubatud.

Näide.

<code>dnsx.sis</code>	<code>dnsx.val</code>
1	LiStS.Ut.Ee
<code>lists.ut.ee</code>	
181	

Dekodeerija

Sisend. Tekstifaili `dnsx.sis` esimesel real on sisendite arv N ($1 \leq N \leq 1000$). Järgmisel N real on igaühel üks kodeerija väljund S_i .

Väljund. Tekstifaili `dnsx.val` väljastada täpselt N rida, igale reale väljastada üks dekodeeritud X_i .

Näide.

<code>dnsx.sis</code>	<code>dnsx.val</code>
1	181
<code>LiStS.Ut.Ee</code>	

Hindamine. Selles ülesandes saab iga lahendus punkte ainult nende testide eest, kus kodeerija ei tee nimes mitte mingeid muid muutusi peale väiketähtede suurtähtedega asendamise ning dekodeerija taastab muudetud nime põhjal õigesti arvu X .

3. Mooduljagamispea

1 sekund 40 punkti

Hammasrataste freesimisel kasutatakse mooduljagamispead, millel on padrun tooriku kinnitamiseks ja vänt padruni liigutamiseks. Vända keeramisel N täisringi võrra teeb padrun täpselt ühe täisringi.

Vända all on plaat, milles on M auguringi, i . ringis on ühtlaste vahedega K_i auku. Vänta on võimalik fikseerida nendesse aukudesse, seega saab vänta lisaks täisringidele keerata ka mingi hulga aukude võrra.

Peaga oli kaasas tabel, kus oli iga võimaliku hammaste arvu H kohta kirjas, mitu täisringi peab vändaga tegema, millist auguringi tuleb kasutada ja mitme augu võrra tuleb vänta veel edasi nihutada lisaks täisringidele, et toorikut järgmise hamba freesimiseks edasi keerata.

Õnnetuseks on see tabel kaotsi läinud ja nüüd on vaja programmi, mis oskaks seda asendada.

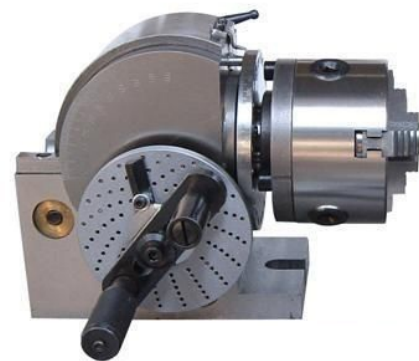
Sisend. Tekstifaili `jaga.sis` esimesel real on N ($1 \leq N \leq 100$), mis näitab, mitu täisringi peab vändaga tegema, et padrun ühe täisringi teeks. Faili teisel real on auguringide arv M ($1 \leq M \leq 10$). Faili kolmandal real on M täisarvu K_i ($1 < K_1 < K_2 < \dots < K_M \leq 100$). Faili neljandal real on vajaliku hammasratta hammaste arv H ($2 \leq H \leq 10\,000$).

Väljund. Kui soovitud hammasratta freesimine selle mooduljagamispeaga ei ole võimalik, väljastada tekstifaili `jaga.val` ainsale reale sõna `EI`. Kui freesimine on võimalik, väljastada faili esimesele reale sõna `JAH` ja teisele reale retsept ühelt hambalt järgmisele liikumiseks. Kui selleks ei ole auguplaati vaja kasutada, väljastada üks täisarv, mis näitab, mitu täisringi tuleb vänta keerata. Kui on vaja kasutada ka auguplaati, väljastada kolm täisarvu, mis näitavad vastavalt, mitu täisringi tuleb vänta keerata, mitme auguga auguringi tuleb kasutada ja mitme augu võrra tuleb vänta lisaks täisringidele edasi liigutada. Kui võimalikke vastuseid on mitu, väljastada ükskõik milline neist.

Näide.	<code>jaga.sis</code>	<code>jaga.val</code>
	8	EI
	2	
	3 4	
	7	

Näide.	<code>jaga.sis</code>	<code>jaga.val</code>
	8	JAH
	2	2
	3 4	
	4	

Näide.	<code>jaga.sis</code>	<code>jaga.val</code>
	8	JAH
	2	0 3 2
	3 4	
	12	



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

1. Sarnased kolmnurgad

1 sekund

30 punkti

Kaht kolmnurka nimetatakse sarnasteks, kui nende külgede vahel on võimalik tekitada selline vastavus, et

- esimese kolmnurga igale küljele vastab täpselt üks teise kolmnurga külj;
- teise kolmnurga igale küljele vastab täpselt üks esimese kolmnurga külj;
- leidub selline konstant k , et esimese kolmnurga iga külje ja talle vastava teise kolmnurga külje pikkuste suhe on täpselt k .

Kirjutada programm, mis kontrollib, kas kaks oma küljepikkustega antud kolmnurka on sarnased.

Sisend. Tekstifaili `kolm.sis` esimesel real on kolm tühikutega eraldatud positiivset täisarvu a_1 , b_1 , c_1 : esimese kolmnurga küljepikkused. Faili teisel real on samuti kolm tühikutega eraldatud positiivset täisarvu a_2 , b_2 , c_2 : teise kolmnurga küljepikkused. Võib eeldada, et ükski küljepikkus ei ületa 10 000.

Väljund. Kui sisendis antud kolmnurgad ei ole sarnased, väljastada tekstifaili `kolm.val` ainsale reale sõna `EI`. Kui kolmnurgad on sarnased, väljastada faili esimesele reale sõna `JAH` ja teisele reale kolmnurkade külgede vaheline vastavus alloleva näite eeskujul kolme murru võrdusena, kus lugejate hulgas esinevad esimese ja nimetaajate hulgas teise kolmnurga küljed, igaüks täpselt ühe korra. Kui võimalikke vastavusi on mitu, väljastada ükskõik milline neist.

Näide.

<code>kolm.sis</code>	<code>kolm.val</code>
2 3 4	EI
3 3 3	

Näide.

<code>kolm.sis</code>	<code>kolm.val</code>
3 4 5	JAH
10 6 8	$a_1/b_2=b_1/c_2=c_1/a_2$

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

2. Ristsumma

1 sekund

30 punkti

Kirjutada programm, mis leiab antud täisarvu ristsumma, see tähendab selle numbrite summa.

Sisend. Tekstifaili `rist.sis` ainsal real on täisarv, mille absoluutväärtus ei ületa 1 000 000 000.

Väljund. Tekstifaili `rist.val` ainsale reale väljastada sisendis antud arvu ristsumma.

Näide.

<code>rist.sis</code>	<code>rist.val</code>
123	6

$1 + 2 + 3 = 6$.

Näide.

<code>rist.sis</code>	<code>rist.val</code>
-1234	10

$1 + 2 + 3 + 4 = 10$.

3. Sõnade liigitamine

1 sekund 40 punkti

On teada, et eesti keeles kirjutatakse nimed alati suure algustähega, aga muud sõnad algavad suure tähega ainult lause alguses.

Kirjutada programm, mis liigitab antud tekstis esinevad sõnad selle reegli põhjal nimeks, muudeks sõnadeks ja sellisteks, mille ühene liigitamine antud teksti põhjal pole võimalik.

Sisend. Tekstifailis `liik.sis` on kuni 30 rida, igaüks pikkusega kuni 80 märki. Failis võivad esineda suured ja väikesed ladina tähed, tühikud, komad, punktid, küsi- ja hüüumärgid. Punktid ning küsi- ja hüüumärgid on lauselõpetajad, aga komad, tühikud ja reavahetused lihtsalt sõnaeraldajad. Faili viimane rida koosneb ainult punktist.

Väljund. Tekstifaili `liik.val` väljastada täpselt kolm rida. Faili esimesele reale väljastada kõik sisendis esinenud sõnad, mille kohta võib eeltoodud reegli põhjal kindlalt väita, et need on nimed. Faili teisele reale väljastada need sõnad, mille kohta võib kindlalt väita, et need ei ole nimed. Faili kolmandale reale väljastada need sõnad, mille kohta ei saa kindlalt väita kumbagi. Sõnade järjekord ridadel ja nende esitamine suurte või väikeste tähtedega pole oluline, kuid iga sõna tuleb väljastada täpselt üks kord ja sõnad tuleb eraldada üksteisest tühikutega.

Näide.	<code>liik.sis</code>	<code>liik.val</code>
	Kati on Mati sober.	Mati Kati
	Mati on Kati sober ka.	on sober ka nende molema
	Mari on nende molema sober.	Mari
	.	

Näide.	<code>liik.sis</code>	<code>liik.val</code>
	Mari ja Marju	ja leidsid palju
	leidsid palju marju.	Mari Marju
	.	