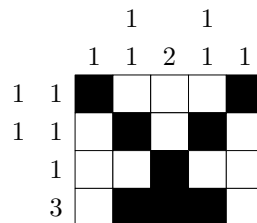


1. Nonogramm (nono)

1 sekund

10 punkti

Nonogramm on mõistatus, mille lahendaja peab taastama ruudustikus mustadest ja valgetest ruutudest moodustatud kujundi. Ruudustiku iga rea ees on arvud, mis näitavad järjestikustest mustadest ruutudest koosnevate ja valgete ruutudega eraldatud plokkide pikkusi ja nende järjekorda. Samamoodi ka veergude kohal. Igal real ja igas veerus peavad näidatud pikkustega mustad plokid olema näidatud järjekorras ja kahe musta ploki vahel peab alati olema vähemalt üks valge ruut.



Kirjutada programm, mis aitab mõistatuse koostajal antud kujundile vastavad ridade ja veergude kirjeldused leida.

Sisend. Tekstifaili `nonosis.txt` esimesel real on kaks täisarvu, ruudustiku ridade arv N ($1 \leq N \leq 100$) ja veergude arv M ($1 \leq M \leq 100$). Järgmisel N real on igaühel täpselt M märki, kus punkt tähistab valget ja trellimärk musta ruutu.

Väljund. Tekstifaili `nonoval.txt` väljastada täpselt $N + M$ rida. Esimesele N reale väljastada ruudustiku ridade ees olevad arvud: ruudustiku esimese rea ees olevad arvud faili reale 1, teise rea ees olevad arvud faili reale 2 j.n.e. Järgmisele M reale väljastada veergude kohal olevad arvud: vasakpoolsseima veeru kohal olevad arvud faili reale $N + 1$, vasakult teise veeru kohal olevad arvud faili reale $N + 2$ j.n.e. Kui ruudustiku mõnes reas või veerus pole ühtki musta ruutu, siis väljastada faili vastavale reale arv 0.

Näide.

<code>nonosis.txt</code>	<code>nonoval.txt</code>
4 5	1 1
#...#	1 1
.#.#.	1
..#..	3
.###.	1
	1 1
	2
	1 1
	1

Näide.

<code>nonosis.txt</code>	<code>nonoval.txt</code>
1 3	1 1
##	1
	0
	1

2. Tehtemärgid (tehe)

1 sekund

20 punkti

Antud jada positiivseid täisarve ning positiivne “vastus”. Panna arvude vahele pluss- ja miinusmärgid nii, et avaldise väärtus oleks etteantud vastus.

Sisend. Tekstifaili `tehehis.txt` esimesel real on jada pikkus N ($1 \leq N \leq 1000$), teisel real jada liikmed ja kolmandal real soovitud tulemus. Ei jada liikmed ega tulemus ületa 10 000.

Väljund. Tekstifaili `teheval.txt` ainsale reale väljastada otsitav avaldis. Võib eeldada, et leidub vähemalt üks lahendus, kus ühegi vahetulemuse aboslutväärtus ei ületa 10 000. Kui võimalikke lahendusi on mitu, väljastada ükskõik milline neist.

Näide.

<code>tehehis.txt</code>	<code>teheval.txt</code>
5	5-4+3+2-1
5 4 3 2 1	
5	

3. Kodufinantsid (fin)

1 sekund

30 punkti

Juku peab oma kulude kohta arvestust tekstifailis, mis on järgmises vormingus (arvepidamine ei tarvite alati 3-tasemeline olla):

```
Märtsi kulud - 1000
  Toit - 500
    Kohukesed - 250
    Liha - 250
  Lõbu - 400
    Pidu - 200
    Kino - 200
  Tervis - 100
```

Järjekordsel faili salvestamisel kaotas tema tekstiredaktor milligipärast kõik taanded ja nüüd näeb fail välja selline:

```
Märtsi kulud - 1000
Toit - 500
Kohukesed - 250
Liha - 250
Lõbu - 400
Pidu - 200
Kino - 200
Tervis - 100
```

Kirjutada programm, mis aitab Jukul arvepidamise taastada, kui on teada, et faili esimene rida on kõigi kulude summa.

Sisend. Tekstifaili `finsis.txt` esimesel real on arvepidamises olnud ridade arv N ($1 \leq N \leq 20$) ja järgmisel N real igaühel ühe kululiigi summa A_i ($1 \leq A_i \leq 1\,000\,000\,000$).

Väljund. Tekstifaili `finval.txt` väljastada täpselt N rida, reale number i sisendfaili real $i + 1$ oleva summa taane. Pange tähele, et esimene taane on alati 0 ja teine taane (kui arvepidamises on rohkem kui üks rida) alati 1. Kui võimalikke vastuseid on mitu, väljastada ükskõik milline neist.

Näide.	finsis.txt	finval.txt
	6	0
	1000	1
	500	2
	250	2
	250	1
	500	2
	500	

4. Miljonär ja vaeslapsed (mort)

1 sek. / 10 sek.

40 punkti

Dickensi-aegsel Inglismaal elas miljonär Mortimer. Temaga samas linnas asusid kolm lastekodu, kus elasid vaeslapsed, kellele Mortimer tavatses jõulukinke teha.

Kinkide jagamise protseduur oli järgmine:

1. Iga vaeslaps saab oma lastekodust korvi, millega ta kingi järele läheb.
2. Mortimer viskab kingitusi järjest laste sekka, mida nood oma korvidega püüavad.
3. Iga kingitus püütakse alati kinni.
4. Lapsed saavad kingitusi kätte juhuslikult, kuid tõenäosus, et konkreetne laps kingituse kätte saab, on võrdeline tema korvisuu pindalaga.
5. Sama lastekodu lastel on sama suurusega korvid.
6. Kui mõni laps saab kingituse kätte, läheb ta sellega kohe lastekodusse tagasi ja rohkem püüdmises ei osale.
7. Lapsi võib olla rohkem kui kingitusi :(

Igal kingitusel on väärtus. Leida iga lastekodu kohta, milline on selle kodu laste poolt saadud kingituste väärtuste keskmine eeldatav summa.

Sisend. Tekstifailis `mortsis.txt` on antud:

1. Esimesel real kaks täisarvu $0 \leq L_1 \leq 100$ ja $1 \leq K_1 \leq 100$, mis tähistavad esimese lastekodu laste arvu ning korvi pindala.
2. Teisel real kaks täisarvu $0 \leq L_2 \leq 100$ ja $1 \leq K_2 \leq 100$, mis tähistavad teise lastekodu laste arvu ning korvi pindala.
3. Kolmandal real kaks täisarvu $0 \leq L_3 \leq 100$ ja $1 \leq K_3 \leq 100$, mis tähistavad kolmanda lastekodu laste arvu ning korvi pindala.
4. Neljandal real on kingituste arv $1 \leq N \leq L_1 + L_2 + L_3$.
5. Ülejäänud N real on kingituste väärtused (täisarvud vahemikus $1 \dots 1000$). Mortimer viskab kingitusi täpselt antud järjekorras.

Väljund. Tekstifaili `mortval.txt` väljastada kolm reaalarvu (täpsusega vähemalt 0,0001), mis vastavad saadud kinkide väärtuste keskmisele eeldatavale summale.

Näide.	mortsis.txt	mortval.txt
	1 1	6.66666666666667
	1 2	11.33333333333333
	1 3	12
	2	
	10	
	20	

Esimese lastekodu ainus laps saab esimese kingituse kätte tõenäosusega $1/6$, see annab oodatavaks väärtuseks $10/6 = 5/3 \approx 1,666667$ ning teiste kingituste püüdmises ta ei osaleks. Tõenäosusega $1/3$ saab esimese kingi teise lastekodu laps ja läheb ära. Sel juhul on esimesel lapsel teise kingi saamise tõenäosus $1/4$. Oodatav väärtus on siis $20/4/3 = 5/3 \approx 1,666667$. Tõenäosusega $1/2$ saab esimese kingi kolmanda lastekodu laps ja läheb ära. Sel juhul on esimesel lapsel teise kingi saamise tõenäosus $1/3$, oodatav väärtus $20/2/3 = 10/3 \approx 3,333333$. Kokku saamegi vastuseks $5/3 + 5/3 + 10/3 = 20/3 \approx 6,666667$. Samasugust arutlust saab kasutada ka teiste laste jaoks.

Näide.	mortsis.txt	mortval.txt
	2 1	27.75
	2 2	51.5
	1 2	25.75
	2	
	63	
	42	

5. Hulkade kahendpuu (puu)

1 sekund

50 punkti

Hulkade kahendpuu erineb tavalisest kahendotsingu puust selle poolest, et puu igas tipus võib olla mitu elementi. Samas kehtib kahendotsingu puu põhitingimuse üldistus: mistahes tipu vasaku alampuu kõik elemendid peavad olema väiksemad ja parema alampuu kõik elemendid suuremad kõigist tipu enda elementidest.

Seega saab hulkade kahendpuus elementi otsida samamoodi kui tavalises kahendotsingu puus. Muuhulgas saab igas tipus otsustada, kas otsitav element peab olema selles tipus või kas tuleks otsingut jätkata vasakus või paremas alampuus. Kui loeme selle otsustamise üheks operatsiooniks, on mistahes elementi sisaldava tipu leidmiseks vaja täpselt nii palju operatsioone, kui palju on tasemeid puu juurest selle elemendi asukohani.

Tippude mäluhalduse iseärasuste tõttu on puu ühes tipus olla võivate elementide maksimaalne arv erinevatel tasemetel erinev. Täpsemalt, kui juurtipu taseme number on 1, siis on tasemel number i ühes tipus olevate elementide maksimaalne arv m_i määratud seostega

$$m_i = \begin{cases} M & \text{kui } i = 1, \\ \max(1, m_{i-1} - D_{((i-1) \bmod K)+1}) & \text{kui } i > 1, \end{cases}$$

kus M , K ja D_j on antud täisarvud ning $(i-1) \bmod K$ on arvu $i-1$ arvuga K jagamisel tekkinud jääk. Näiteks kui $M = 4$, $K = 2$ ja $D_1 = 1$, $D_2 = 2$, siis võib puu juurtipus olla maksimaalselt $m_1 = M = 4$, tema vahetutes alluvates kummaski maksimaalselt $m_2 = m_1 - D_1 = 4 - 1 = 3$, nende vahetutes alluvates igaühes $m_3 = m_2 - D_2 = 3 - 2 = 1$ ja kõigis järgmistes kihtides igas tipus maksimaalselt $m_i = \max(1, \dots) = 1$ element.

Lisaks on teada, et puu kasutamisel otsitakse erinevaid elemente sellest erinevate tõenäosustega ja seega ei tarvitse balanseeritud puu anda parimat keskmist päringule vastamise aega. Näiteks kui ülekaalukalt kõige sagedamini otsitakse minimaalset elementi, on kasulik panna see puu juurtippu ja siis peab kogu vasaku alampuu jääma tühjaks.

Ülesanne on leida keskmine ühe elemendi otsimiseks vajalik operatsioonide arv antud elementide hulga ja päringute tõenäosuste jaoks optimaalse kujuga hulkade kahendpuus.

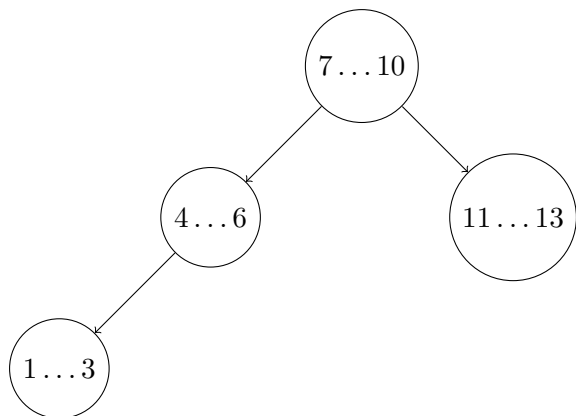
Sisend. Tekstifaili `puusis.txt` esimesel real on kolm täisarvu: hulga elementide arv N ($1 \leq N \leq 100$) ning M ($1 \leq M \leq N$) ja K ($1 \leq K \leq N$). Järgmisel K real on igaühel üks täisarv: real $j+1$ on D_j väärtus ($0 \leq D_j \leq M$). Faili viimasel real on N reaalarvu P_i ($0 \leq P_i \leq 1$; $\sum_{i=1}^N P_i = 1$): elementide otsimise tõenäosused. Tõenäosused on järjestatud elementide väärtuste järgi (esimene tõenäosus vastab vähimale elemendile). Pange tähele, et elementide väärtused ei ole lahenduse jaoks olulised; võib eeldada, et nad on kõik erinevad.

Väljund. Tekstifaili `puuval.txt` ainsale reale väljasta üks reaalarv, ühele otsingule kuluv keskmine operatsioonide arv optimaalse kujuga puus. Väljastatud vastus võib täpsest erineda ülimalt 10^{-6} võrra.

Näide.	<code>puusis.txt</code>	<code>puuval.txt</code>
	4 2 1	1.5000000
	2	
	0.2 0.2 0.3 0.3	

Näide. puusis.txt puuval.txt
 13 4 2 1.7200000
 1
 0
 0.06 0.06 0.06 0.06 0.06 0.06 0.115 0.115 0.115 0.115 0.06 0.06 0.06

Teises näites on optimaalne puu kuju järgmine:



6. Perimeeter (per)

50 punkti

Informaatikaolümpiaadi ettevalmistuslaagris anti osalejatele lahendada järgmine ülesanne:

Ristkülikute ühendi perimeeter

Tasandil on antud N ristkülikut, mille servad on koordinaattelgedega paralleelsed. Leida nende ristkülikute ühendi perimeeter ehk välise kontuuri pikkus. Tekkinud kujundi sees olevaid auke ei ole vaja arvesse võtta, kuid ühe ristkülikutest koosneva kujundi augus asuvat teist ristkülikutest koosnevat kujundit, mis väliskujundit ei puuduta üheski punktis tuleb lugeda ikkagi omaette kujundiks.

Sisend. Faili esimesel real on antud ristkülikute arv N ($1 \leq N \leq 1000$). Järgmisel N real on igaühel neli reaalarvu LX , LY , UX ja UY , kus LX ja LY on ristküliku alumise vasaku ning UX ja UY ülemise parema nurga koordinaadid. Koordinaatide väärtused on 0 kuni 1 000 000, mille esituseks piisab 32-bitisest `float` arvutüübist.

Väljund. Faili ainsale reale väljastada leitud ristkülikute ühendi perimeeter täpsusega kuni kolm kohta pärast koma.

Näide.	Sisendfail	Väljundfail
	3	564.0
	7.0 170.0 99.5 190.0	
	0.5 100.0 12.0 225.0	
	10.0 80.0 50.0 110.0	

Teie ülesandeks on koostada testid, mis kontrolliksid esitatud lahenduste korrektsust.

Hindamine. Kõik testandmed koondada ühte tekstifaili järgnevalt kirjeldatud formaadis. Faili esimesel real on testide arv K ($1 \leq K \leq 20$). Järgnevas K blokis on bloki esimesel real oodatav tulemus ning selle järel sisendandmed samal kujul kui ülesande kirjelduses. Lahendusena tuleb esitada see tekstifail.

Näide.	Väljundfail
	2
	4.0
	1
	1.0 1.0 2.0 2.0
	564.0
	3
	7.0 170.0 99.5 190.0
	0.5 100.0 12.0 225.0
	10.0 80.0 50.0 110.0

Hindamine. Esitatud testidega testitakse kümmet programmi, mis teadaolevat sisaldavad igaüks vähemalt üht viga, kuid lahendavad õigesti näites esitatud testjuhud. Iga programm, mille puhul esitatud testid tuvastavad vea olemasolu, annab 5 punkti.

7. Paroolide tuvastamine (pass)

50 punkti

Tarkvarafirma ABCDEF-software on loonud veebipõhise kasutajate haldamise mooduli, mille kohta on teada:

- kasutajanimi ja parool võivad olla 1...20 märki pikad ning koosneda sümbolitest A...Z, a...z, 0...9 (täpitähed pole lubatud);
- kasutajanimed peavad olema unikaalsed;
- nii kasutajanimed kui paroolid on tõstutundlikud.

Teile on kättesaadavad selle mooduli kaks funktsiooni:

- kõigi kasutajanimede nimekirja kuvamine;
- uue kasutaja loomine (saate määrata nii kasutajanime kui parooli).

Moodulisse on juba sisestatud mõned kasutajad ja vaja on tuvastada nende paroolid.

Hindamine. Lahendusena esitada tekstifail, mille igal real on üks kasutajanimi ja selle järel tühikuga eraldatult kasutaja parool. Hindamisel arvestatakse nii tuvastatud paroolide arvu kui süsteemi loodud uute kasutajate hulka:

1. Kõik lahendused, mis on õigesti tuvastanud vähemalt ühe parooli, järjestatakse tuvastatud paroolide arvu alusel.
2. Võrdse hulga paroolide tuvastanud lahenduste hulgas on eespool lahendus, mis on teinud süsteemi vähem uusi kasutajaid.
3. Pingereas esimene lahendus saab 50 punkti ja viimane 5 punkti; ülejäänud lahendused saavad punkte proportsionaalselt nende kohaga pingereas.

Märkus. Selle ülesande sisuks on paroolide äraarvamine kahe eeltoodud funktsiooni kasutamise teel, mitte serveri ründamine mingil muul moel. Lubatud (ja isegi soovitatav) on kirjutada programme, mis teostavad automatiseeritud päringuid mooduli funktsioonidele, kuid see ei ole kohustuslik. Tuleks arvestada, et väga suure hulga kasutajate loomise järel võib moodul muutuda aeglasemaks või lõpetada üldse töö.

Kasutajate haldamise mooduli kasutamine

Kasutajate haldamise moodulile saab ligi aadressil <http://cms.eio.ut.ee/pass/>.

Eduka sisselogimise järel EIO serveri kasutajanime ning parooliga annab moodul brauserile küpsise (<https://et.wikipedia.org/wiki/HTTP-küpsis>). Küpsis kehtib kogu võistluse. (Kui soovite lahendada ülesannet programmi abil, pole tarvis realiseerida EIO kasutajatunnusega sisselogimist, piisab kui lisate oma programmi saadetavale päringule küpsised.)

Moodulist on teile kasutada kaks versiooni:

1. <http://cms.eio.ut.ee/pass/test.cgi>

See versioon sisaldab täiendavat lähtestusfunktsiooni, mis kustutab kõik teie loodud kasutajad ning taastab mooduli algse seisu. See versioon on mõeldud oma lahenduse testimiseks, punkte selle lahendamise eest ei anta.

Selle mooduli kasutajate andmebaasi sisu on:

```
'adam','eve'  
'joe','hacker1'  
'bill','1234'
```

2. <http://cms.eio.ut.ee/pass/live.cgi>

Selles versioonis lähtestusfunktsiooni ei ole ning igale võistlejale genereeritakse sarnase keerukusega, kuid erinev kasutajate andmebaas. Hinnatakse ainult selle alamülesande vastust.

Mooduli poolt aktsepteeritavad parameetrid:

- **q=list** — kuvab baasis olevate kasutajate nimikirja; täiendavaid parameetreid pole;
- **q=reset** — lähtestab andmebaasi; täiendavaid parameetreid pole;
- **q=adduser** — loob uue kasutaja; täiendavad parameetrid: **user=**, **pass1=**, **pass2=** (loodava kasutaja nimi ning parool kaks korda, nagu ikka parooli valimise vormidel).

Mooduli poole võib pöörduda nii GET kui POST meetodiga. Moodul töötleb ainult autoriseeritud päringuid (nõutav korrektse küpsise olemasolu).