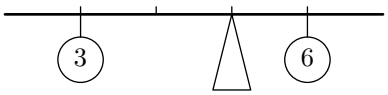
1. Kang

1 sekund 10 punkti

Vaatleme kangi pikkusega L, mille külge on riputatud N raskust. Iga raskuse kohta on teada tema riputuskoht X_i (riputuskoha kaugus kangi vasakust otsast) ja mass M_i.

Kirjutada programm, mis leiab, kuhu (kui kaugele kangi vasakust otsast) tuleb panna tugi, et kang oleks sellel tasakaalus, kui kangi enda mass on null.



Sisend. Tekstifaili `kangsis.txt` esimesel real on kaks tühikuga eraldatud täisarvu: kangi pikkus L ($0 < L \leqslant 100$) ja raskuste arv N ($1 \leqslant N \leqslant 100$). Järgmisel N real on igaühel kaks tühikuga eraldatud täisarvu: raskuse i riputuskoht X_i ($0 \leqslant X_i \leqslant L$) ja mass M_i ($0 < M_i \leqslant 100$).

Väljund. Tekstifaili `kangval.txt` ainsale reale väljastada üks reaalarv: toetuspunkti asukoht (selle kaugus kangi vasakust otsast). Väljastatud väärtus ei tohi täpsest vastusest erineda rohkem kui 0,01 võrra.

Näide.

`kangsis.txt`
```
5 2
1 3
4 6
```

`kangval.txt`
```
3.0
```

Hindamine. Testides koguväärtusega 5 punkti kehtib lisaks tingimus $N \leqslant 5$.

2. Kaartide sorteerimine

1 sekund 20 punkti

Kaardimänguseisude märkimiseks kasutatakse enamasti tähistust, kus iga kaart on antud kahe märgina, mis näitavad kaardi kõrgust ja masti. Kõrguste tähised on `AKQJT98765432` (*ace, king, queen, jack, ten, 9, . . . , 2* — äss, kuningas, emand, soldat, 10, 9, . . . , 2) ja mastide omad `shdc` (*spades, hearts, diamonds, clubs* — poti, ärtu, ruutu, risti). Näiteks poti emanda tähis on `Qs`.

Kirjutada programm, mis järjestab antud kaardid kõigepealt masti ning seejärel sama masti kaardid omavahel kõrguse järgi (eelmises lõigus antud järjekorras).

Sisend. Tekstifaili `sortsis.txt` esimesel real on kaartide arv N ($1 \leqslant N \leqslant 52$) ja teisel real $2 \cdot N$ märgist koosnev sõne, mis kirjeldab N kaarti (ükski kaart ei kordu).

Väljund. Tekstifaili `sortval.txt` ainsale reale väljastada $2 \cdot N$ märgist koosnev sõne, mis kirjeldab sisendis antud kaarte nõutud järjekorras.

Näide.

`sortsis.txt`
```
5
3d8sAdTd6c
```

`sortval.txt`
```
8sAdTd3d6c
```

3. Hanoi torn

1 sekund 30 punkti

Hanoi torn on nuputusmäng, mis koosneb kolmest vardast ja N kettast. Ketaste läbimõõdud on $1 \ldots N$ ja iga läbimõõduga kettaid on täpselt üks. Iga ketta keskel on auk, millest varras läbi mahub. Algseisus on kõik kettad suuruse järjekorras vasakpoolsel vardal ja mängija eesmärk on paigutada nad ümber parempoolsele vardale, kasutades keskmist varrast ajutise hoiukohana. Seejuures tohib igal sammul tõsta ühe varda kõige pealmise ketta mingile teisele vardale kõige pealmiseks kettaks ja ühelgi vardal ei tohi panna suuremat ketast väiksema peale.

Kirjutada programm, mis loeb sisse mingi mänguseisu ja leiab vähima tõstmiste arvu, millega on võimalik mäng sellest seisust lõpuni mängida (s.t viia kõik kettad parempoolsele vardale õigesse järjekorda).

Sisend. Tekstifaili `tornsis.txt` esimesel real on kõigepealt vasakul vardal olevate ketaste arv N_v ja seejärel N_v tühikutega eraldatud täisarvu: ketaste läbimõõdud loetletuna altpoolt ülespoole; faili teisel real on samal kujul keskmise varda ketaste arv N_k ja nende ketaste mõõdud ning kolmandal real parema varda ketaste arv N_p ja nende mõõdud ($0 \leqslant N_v$, $0 \leqslant N_k$, $0 \leqslant N_p$, $N_v + N_k + N_p \leqslant 30$). Võib eeldada, et seis vastab mängureeglitele (iga ketas $1 \ldots N_v + N_p + N_k$ esineb täpselt ühe korra ja kuskil pole suurem ketas väiksema peal).

Väljund. Tekstifaili `tornval.txt` ainsale reale väljastada üks täisarv: minimaalne sammude arv, millega on võimalik mäng sisendis kirjeldatud seisust lõpuni mängida.

Näide.

`tornsis.txt`
```
2 2 1
1 3
0
```

`tornval.txt`
```
4
```

Sisendis toodud seisu saab nelja tõstega lõpuni mängida nii: ketas 3 keskmiselt vardalt paremale; ketas 1 vasakult vardalt keskmisele; ketas 2 vasakult vardalt paremale; ketas 1 keskmiselt vardalt paremale. Kolme sammuga mängu lõpetada ei saa, sest siis peaks iga ketta kohe esimese tõstega paremale vardale viima; see aga pole võimalik, sest siis jääks ketas 1 ketta 2 alla.

Hindamine. Testides koguväärtusega 20 punkti kehtib lisaks tingimus $N_v + N_k + N_p \leqslant 10$.

4. Ristsumma

1 sekund 40 punkti

Naturaalarvu ristsummaks nimetatakse selle numbrite summat. Näiteks arvu 123 ristsumma on $1 + 2 + 3 = 6$ ja arvu 99 ristsumma $9 + 9 = 18$.

Kirjutada programm, mis leiab, kui palju on selliseid antud arvust väiksemaid naturaalarve, mille ristsumma on võrdne antud arvu ristsummaga.

Sisend. Tekstifaili `ristsis.txt` ainsal real on täisarv N ($0 \leqslant N \leqslant 10^{18}$).

Väljund. Tekstifaili `ristval.txt` ainsale reale väljastada selliste arvust N väiksemate naturaalarvude arv, mille ristsumma on võrdne arvu N ristsummaga.

Näide.

`ristsis.txt`
```
123
```

`ristval.txt`
```
9
```

Loendatavad arvud on 6, 15, 24, 33, 42, 51, 60, 105, 114.

Hindamine. Testides koguväärtusega 20 punkti kehtib lisaks tingimus $N \leqslant 10^9$ ja nende hulgas testides koguväärtusega 10 punkti lisaks veel $N \leqslant 10^3$.

5. Sallide söömine

5 sekundit 50 punkti

Moekunstnik Märdil on kapis N ilusat salli, millega ta käib laupäeviti moekunstnike koosolekul. Igal sallil on oma moeväärtus ja koosolekul saab Märt vastava hulga feimi. Kaks korda sama salliga kohale tulla oleks suur *faux pas* ja Märt ei tee seda kunagi. Kantud sallid paneb ta kappi tagasi, aga rohkem neid ei kanna.

Märdi kapis elab ka koiliblikas Kärt, kes sööb igal pühapäeval ühele sallile augu sisse. Auguga salli enam kanda ei saa. Kärt moeväärtusest ei hooli, vaid sööb salle juhuslikult. Mingile sallile augu söömise tõenäosus on võrdeline salli pikkusega. Kärt võib sama salli süüa ka mitu korda.

On selge, et Märt saaks koosolekul käia ülimalt N korda, kui Kärt sööks ainult juba kantud salle. Kui Kärt sööb mõnikord ka kandmata salle, jääb koosolekute arv sellevõrra väiksemaks.

Kirjutada programm, mis leiab, kui palju Märt keskmiselt feimi saab, kui ta kasutab parimat strateegiat, aga Kärt sööb salle juhuslikult. Tegevus algab nädala alguses, seega esimese koosoleku ajaks on kõik sallid veel terved.

Sisend. Tekstifaili `sallsis.txt` esimesel real on sallide arv N ($1 \leqslant N \leqslant 20$). Järgmisel N real on igaühel kaks täisarvu M_i ja P_i ($1 \leqslant M_i \leqslant 100$, $1 \leqslant P_i \leqslant 20$): ühe salli moeväärtus ja pikkus.

Väljund. Tekstifaili `sallval.txt` ainsale reale väljastada üks reaalarv: Märdi keskmine feim, kui ta valib sallide kandmise järjekorra optimaalselt, aga Kärt sööb neid juhuslikult. Väljastatud väärtus ei tohi täpsest vastusest erineda rohkem kui 0,001 võrra.

Näide.

`sallsis.txt`
```
3
10 3
20 2
30 1
```

`sallval.txt`
```
48.333
```

Selle sisendi puhul on Märdi optimaalne strateegia võtta kõigepealt teine sall. Pärast seda on:

- tõenäosusega $\frac{1}{2}$ auk esimeses sallis, teiseks koosolekuks alles ainult kolmas sall ja saame kokku 50 feimi;
- tõenäosusega $\frac{1}{3}$ auk teises sallis ja kaks salli alles; võtame neist kõigepealt kolmanda ning $\frac{1}{2}$ tõenäosusega saame ka esimese ära kanda; keskmiselt kokku 55 feimi;
- tõenäosusega $\frac{1}{6}$ auk kolmandas sallis, alles ainult esimene ja saame kokku 30 feimi.

Nende variantide kaalutud keskmine ongi $48\frac{1}{3}$ feimi.

Näide.

`sallsis.txt`
```
4
10 10
20 9
30 5
40 1
```

`sallval.txt`
```
84.94144
```

Hindamine. Testides koguväärtusega 25 punkti kehtib lisaks tingimus $N \leqslant 10$.

6. Pindala

Testimine 50 punkti

Inseneribüroo tellis tarkvarafirmalt programmi, mis peab oskama arvutada kolmnurga ja nelinurga ühisosa pindala. Programm peab oskama arvutada mistahes kujuga kolm- ja nelinurkade ühisosasid.

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

Sisend. Tekstifaili `pindsis.txt` esimesel real on kuus tühikutega eraldatud täisarvu A_x, A_y, B_x, B_y, C_x, C_y: kolmnurga ABC tippude koordinaadid. Faili teisel real on sarnaselt nelinurga $DEFG$ tippude koordinaadid tippude nelinurga piirjoonel esinemise järjekorras. Kõik koordinaadid on täisarvud, mille absoluutväärtused ei ületa 100.

Väljund. Tekstifaili `pindval.txt` ainsal real on üks reaalarv: sisendis kirjeldatud kujundite ühisosa pindala, mis ei tohi täpsest vastusest erineda rohkem kui 0,01 võrra.

Näide.

`pindsis.txt`
```
1 1 5 1 3 5
1 3 5 3 5 6 1 6
```

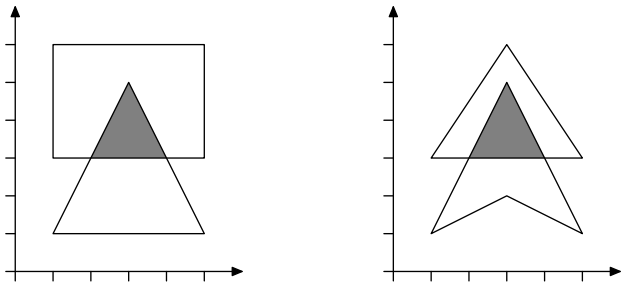
`pindval.txt`
```
2.0
```

Näide.

`pindsis.txt`
```
1 3 5 3 3 6
1 1 3 2 5 1 3 5
```

`pindval.txt`
```
2.0
```

Alloleval joonisel on vasakul kujutatud esimese ja paremal teise näite sisendandmed.



Hindamine. Selles ülesandes tuleb lahendusena esitada üks ZIP-fail, mille sees on kuni 15 sisendfaili nimedega `pindsis01.txt`, `pindsis02.txt`, ... ja neile vastavad väljundfailid nimedega `pindval01.txt`, `pindval02.txt`, ...

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

7. Must kast

1 sekund 75 punkti

Andmete sisestamisel, edastamisel ja salvestamisel tekkida võivate vigade tuvastamiseks kasutatakse sageli kontrollsummasid. Näiteks kui saatja tahab edastada sõnumi M, arvutab ta sellest mingi varem kokkulepitud funktsiooni f väärtuse $S = f(M)$ ja edastab sõnumi M asemel paari $\langle M, S \rangle$. Kui vastuvõtja saab sidekanalist paari $\langle M', S' \rangle$, kontrollib ta, kas $f(M') = S'$. Kui see võrdus ei kehti, siis on kindlasti saatmisel mingi viga tekkinud ja tuleks paluda saatjal sõnum uuesti saata.

Kui võrdus kehtib, siis ei ole see veel päris täielik garantii, et andmed on tervelt kohale jõudnud. Nimelt on võimalik, et sidekanalis on riknenud nii M kui ka S väärtus ja teinud seda (kas juhuslikult või kellegi kurja kavatsuse tõttu) nii, et vastuvõtja tehtav kontroll seda ei tuvasta. Sellise valepositiivse tulemuse saamise tõenäosust on võimalik vähendada funktsiooni f sobiva valimisega. Selles ülesandes vaatleme erinevaid kontrollsummade arvutamiseks kasutatavaid funktsioonide tüüpe.

Paarsusbitt.[1] Kõige lihtsam kontrollsumma on 1-bitine ja selle biti väärtus valitakse vastavalt edastatavate andmete kahendkujus olevate 1-bittide arvule. Aja jooksul on kasutatud selle skeemi mitmeid variante: mõnes nõutakse, et sõnumis ja kontrollsummas kokku peab olema paarisarv 1-bitte, mõnes, et paaritu arv; mõnes lisatakse kontrollbitt edastatavate andmete lõppu, mõnes algusesse. Kuna selles skeemis on kontrollsummal ainult kaks võimalikku väärtust, on valepositiivse tõenäosus üsna suur, tervelt 50%. Vastukaaluks on sellist kontrollsummat väga lihtne elektrooniliselt realiseerida.

Ristsumma.[2] Paarsusbitist natuke veakindlam skeem on edastada andmetega koos nende ristsumma. Näiteks kui sidekanali rikke tõttu edastatakse iga biti tegeliku väärtuse asemel 0-bitt ja selline veaolukord kestab parasjagu nii kaua, et 0-bittidega asendatakse paarisarv esialgses sõnumis olnud 1-bitte, siis paarsusbitt seda viga ei tuvasta, aga edastatud andmete ristsumma muutub sellise veaga kindlasti väiksemaks.

Kaalutud ristsumma.[3] Lihtne ristsumma ei tuvasta sõnumis märkide järjekorra muutumist (mis võib kergesti juhtuda näiteks siis, kui inimene andmeid sisestab). Selliste vigade tuvastamiseks lisatakse näiteks pankades konto- ja viitenumbritele kontrolljärk nii, et kui korrutada tulemuses järkude väärtusi enne nende summeerimist kaaludega $3, 7, 1, 3, 7, 1, \ldots$, siis korrektse konto- või viitenumbri korral on summa alati 10 kordne.

CRC.[4] Eelmistest märksa keerulisemad on CRC-summad. Nende arvutamisel vaadeldakse bitte kui polünoomide kordajaid ja kontrollsumma arvutatakse kui edastatava sõnumi bittidega määratud polünoomi P ja mingi fikseeritud polünoomi G jagamisel tekkiv jääk (nagu täisarvude P ja G puhul on võimalik leida jagatis Q ja jääk R nii, et $P = G \cdot Q + R$, saab sama teha ka polünoomidega; tasub tähele panna, et nende polünoomide kordajaid käsitletakse arvutamisel ühebitiste täisarvudena, mille liitmisel ületäitumise tõttu $1 + 1 = 0$). Sellestki skeemist on aja jooksul kasutusel olnud hulk erinevaid variatsioone erinevate kontrollsummade pikkuste ja erinevate jagajatega, samuti on varieerunud nii ühes baidis bittide kui ka mitmebaidises kontrollsummas baitide lugemise järjekord.

Krüptograafilised räsifunktsioonid.[5] Eelnevalt kirjeldatud kontrollsummad on mõeldud juhuslike vigade tuvastamiseks ja ei kaitse tahtliku ründe vastu. Tõepoolest, ründaja võib

[1] http://en.wikipedia.org/wiki/Parity_bit
[2] http://en.wikipedia.org/wiki/Digit_sum
[3] http://en.wikipedia.org/wiki/Check_digit
[4] http://en.wikipedia.org/wiki/Cyclic_redundancy_check
[5] http://en.wikipedia.org/wiki/Cryptographic_hash_function

pärast sõnumi muutmist uue kontrollsumma arvutada ja edastada koos muudetud sõnumiga ka muudetud kontrollsumma. Selle välistamiseks peab kontrollsumma skeemil olema kaks lisaomadust.

Esiteks tuleb kasutada sellist funktsiooni f, mille puhul on raske sõnumit niimoodi muuta, et muudetud sõnumi kontrollsumma on sama kui esialgse oma. Krüptograafilised räsifunktsioonid, millest tuntuimad on MD5, SHA1 ja SHA2 perekond, on just sellised.

Teiseks peab saatjal ja vastuvõtjal olema salasõna, mida ründaja ei tea. Kõige lihtsamal juhul võib salasõna P kasutada nii, et $f(M)$ asemel arvutab saatja kontrollsumma kujul $S = f(MP)$ (kontrollsumma arvutamiseks lisatakse salasõna P sõnumi M lõppu), aga edastab ikka ainult $\langle M, S\rangle$. Kui nüüd ründaja tahab sõnumi M asendada sõnumiga M', siis ei saa ta seda teha, sest ta ei tea salasõna P ja ei saa seega arvutada $f(M'P)$ väärtust. Kui sõnumi vastuvõtja salasõna teab, siis tema saab vastuvõetud sõnumist ja talle teadaolevast salasõnast kontrollsumma uuesti arvutada ja selle vastavust kontrollida.

Aadresil `http://prog.offline.ee/html2/kast.cgi` on viis veebirakendust, mis arvutavad eelpool kirjeldatud skeemide järgi kontrollsummasid. Leida iga rakenduse kohta, millise skeemi millist varianti (ja räsifunktsiooniga skeemi korral ka, millist parooli) see rakendus kasutab ja kirjutada programm, mis arvutab täpselt samasuguseid kontrollsummasid.

Hindamine. Iga veebirakendus on eraldi alamülesanne. Iga alamülesande lahendusena tuleb esitada eraldi programm ja iga programmi hinnatakse eraldi. Programme testitakse failide kaudu sisendi-väljundiga, oma lahendustele veebiliideseid mitte teha.

Sisend. Tekstifaili `kastsis.txt` esimesel real on sõnumite arv N ($1 \leqslant N \leqslant 100$) ja järgmisel N real igaühel üks sõnum. Võib eeldada, et

- kõik sõnumid koosnevad 7-bitise ASCII kooditabeli trükitavatest märkidest;
- ühegi sõnumi pikkus ei ületa 80 märki;
- testimiseks kasutatakse ainult sõnumeid, mida vastav veebiteenus aktsepteerib.

Väljund. Tekstifaili `kastval.txt` väljastada täpselt N rida. Väljundi i. reale väljastada sisendi real $i + 1$ olnud sõnum koos kontrollsummaga täpselt samas vormingus, mida kasutab vastav veebirakendus.

Näide.

`kastsis.txt`	`kastval.txt`
`2`	`246`
`123`	`912`
`456`	