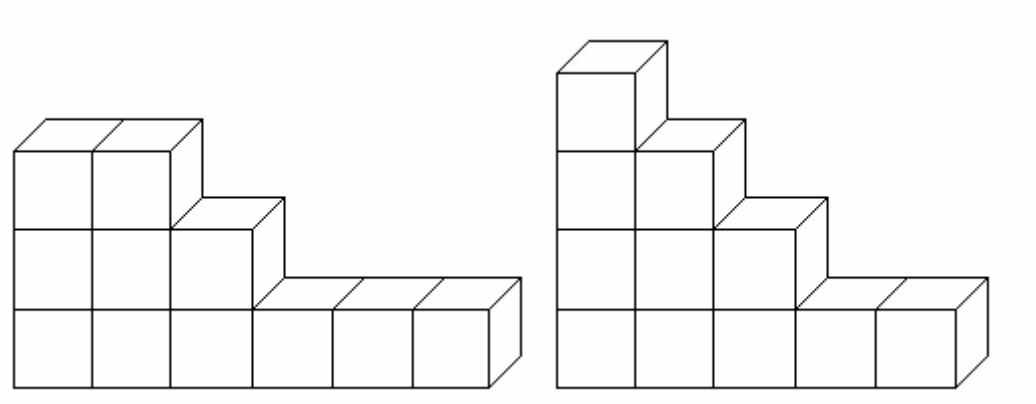
LATVIJAS REPUBLIKAS 20. INFORMĀTIKAS OLIMPIĀDES IETEICAMIE I POSMA UZDEVUMI

1. ”TREPĪTES”

Doti N vienādi kubiņi, no kuriem jāizveido *trepītes*, ievērojot sekojošus nosacījumus:

- Kubiņus liek ar sānu skaldnēm cieši vienu pie otra blakus, veidojot *rindas*,
- Kubiņu rindas liek vienu virs otras, izlīdzinot pēc kreisās malas - t.i. visu rindu pirmie kubiņi atrodas tieši viens virs otra,
- Katrā nākošajā (augstāk esošā) rindā ir mazāk kubiņu nekā iepriekšējā,
- Ir vismaz divas kubiņu rindas,
- Ir jāizmanto visi kubiņi.



Zīmējumā parādītas divas trepītes no 11 kubiņiem. Aprēķiniet, cik dažādas trepītes var izveidot no N kubiņiem!

Ievaddati

Teksta faila `trepites.in` vienīgajā rindā dots naturāls skaitlis N ($1 \leq N \leq 750$).

Izvaddati

Teksta faila `trepites.out` vienīgajā rindā jāizvada vesels skaitlis – dažādo trepīšu skaits.

Piemēri

Ievaddati (fails `trepites.in`)	Izvaddati (fails `trepites.out`)
3	1
1	0
6	3
11	11
100	444792

2. ”PRETESTĪBAS”

Pēteris Neuzmanīgais bija ļoti cītīgi sagatavojies fizikas kontroldarbam un iemācījies visu vajadzīgo teoriju, taču pieļāva neskaitāmas neuzmanības kļūdas un tādēļ saņēma nesekmīgu atzīmi. Skolotāja Sliktere ļāva Pēterim uzlabot atzīmi, izpildot papildus mājasdarbu, kurā Pēterim jāparāda savas zināšanas dažādu slēgumu kopējās pretestības noteikšanā. Pēteris ir izpildījis mājasdarbu, taču baidās, ka atkal varētu būt pieļāvis neuzmanības kļūdas, tāpēc viņš lūdz uzrakstīt programmu, kas izrēķinātu doto slēgumu pretestību, lai viņš varētu tās iegūtos rezultātus salīdzināt ar savējiem.

Pēteris zin, ka pretestības mēra omos un to, ka, ja divas pretestības vai divus jau izveidotus slēgumus saslēdz virknē, tad to kopējo pretestību aprēķina saskaitot abu saslēdzamo pretestību vai slēgumu pretestības: $R_k = R_1 + R_2$. Savukārt, ja divas pretestības vai jau izveidotus slēgumus saslēdz paralēli, tad spēkā ir vienādība $1/R_k = 1/R_1 + 1/R_2$.

Jūsu uzdevums – uzrakstīt datorprogrammu, kas aprēķina doto slēgumu pretestības.

Ievaddati

Teksta faila `rezistor.in` katra no rindiņām var būt kāda no sekojošiem tipiem:

1) Pirmais simbols-identifikators ir “R”, pēc kura seko atstarpe un viens vesels skaitlis S ($1 \le S \le 1000000$). Šāda tipa rindiņa apraksta slēgumu, kuru veido viens pats rezistors, kura pretestība ir S omi.

2) Pirmais simbols-identifikators ir “V”, pēc kura seko divi ar atstarpi atdalīti veseli skaitļi N_1 un N_2 ($1 \le N_1, N_2 < N_c$), kur N_c ir šīs rindiņas numurs ievad failā. Šāda tipa rindiņa apraksta slēgumu, kuru iegūst virknē saslēdzot N_1-ajā un N_2-ajā rindiņā aprakstīto slēgumu kopijas.

3) Pirmais simbols-identifikators ir “P”, pēc kura seko divi ar atstarpi atdalīti veseli skaitļi N_1 un N_2 ($1 \le N_1, N_2 < N_c$), kur N_c ir šīs rindiņas numurs ievad failā. Šāda tipa rindiņa apraksta slēgumu, kuru iegūst paralēli saslēdzot N_1-ajā un N_2-ajā rindiņā aprakstīto slēgumu kopijas.

4) Pirmais un vienīgais simbols-identifikators ir “X”. Šī rindiņa apzīmē faila beigas un ir pēdējā rindiņa ievad failā.

Ievaddatu failā būs ne vairāk kā 100001 rindiņa.

Izvaddati

Teksta failam `rezistor.out` ir jāsatur par vienu rindiņu mazāk nekā ievad failam, katrā no tām jāatrodas vienam pozitīvam skaitlim – izvaddatu faila i-tajā rindiņā jāatrodas ievaddatu faila i-ajā rindiņā aprakstītā slēguma pretestībai omos. Rezultāti jāizvada ar tieši divām zīmēm aiz decimālā punkta, noapaļojot līdz simtdaļām. Zināms, ka neviena slēguma pretestība nebūs lielāka par 10000000 omiem.

Piemērs

Ievaddati (fails `rezistor.in`)	Izvaddati (fails `rezistor.out`)
R 2 P 1 1 P 1 2 V 3 3 V 1 3 X	2.00 1.00 0.67 1.33 2.67

3."ROMIEŠU SKAITĻI"

Lai naturālu skaitli pierakstītu romiešu skaitīšanas sistēmā, tiek izmantoti septiņi latīņu alfabēta lielie burti, pie kam katrs no šiem burtiem apzīmē noteiktu skaitlisku vērtību:

Burts	Vērtība
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Uzrakstot no šiem burtiem vienu vai vairākus burtus pēc kārtas, var apzīmēt jebkuru naturālu skaitli no 1 līdz 3999. Romiešu skaitļu veidošanas likumi nav īsi aprakstāmi, bet, ja zināms, ka dotā burtu virkne **ir** korekts skaitlis romiešu skaitīšanas sistēmā, tad tā skaitliskās vērtības iegūšanu var aprakstīt šādi:

Skaitļa vērtību nosaka visu burtu vērtību summa, kur katra burta vērtība ir ņemta vai nu ar '+' vai '-' zīmi. Ja aiz burta virknē seko cits burts ar lielāku vērtību, tad šī burta vērtība jāņem ar '-' zīmi, bet visos citos gadījumos – ar '+' zīmi.

Piemēram, skaitļa MCMXIV skaitliskā vērtība ir `1000-100+1000+10-1+5=1914`, bet skaitļa MMVII vērtība ir `2007`.

Uzrakstiet programmu, kas ievadītam naturāla skaitļa pierakstam romiešu skaitīšanas sistēmā izvada tā skaitlisko vērtību decimālā skaitīšanas sistēmā!

Ievaddati

Teksta faila `romiesi.dat` vienīgajā rindā dota burtu virkne bez atdalošiem tukšumiem. Virknē var būt tikai lielie burti I,V,X,L,C,D un M. Ir zināms, ka dotā virkne ir korekts naturāla skaitļa pieraksts romiešu skaitīšanas sistēmā un tās garums ir robežās no 1 līdz 15.

Izvaddati

Teksta faila `romiesi.rez` vienīgajā rindā jāizvada naturāls skaitlis - ievadītā skaitļa vērtība decimālā skaitīšanas sistēmā.

Piemēri

Ievaddati (`romiesi.dat`)	Izvaddati (`romiesi.rez`)
`MCMXIV`	`1914`
`V`	`5`
`DIV`	`504`
`MMVII`	`2007`

4.”PĀRPROTAMIE DATUMI"

Lasot datumu, kas apzīmē kādu dienu XXI gadsimtā un kas pierakstīts formā xx/yy/zz – kur viens burts apzīmē tieši vienu ciparu, bet nav zināms, kurš tieši skaitlis (xx vai yy, vai zz) apzīmē gada pēdējos divus ciparus, kurš mēnesi un kurš mēneša dienu, iespējamas divdomības, cenšoties interpretēt šo datumu.

Piemēram, 86/12/24 viennozīmīgi apzīmē 2086. gada 24. decembri, jo ir skaidrs, ka 86 var apzīmēt tikai gadu (nevienā mēnesī nav 86 dienu un gadā nav 86 mēnešu), 24 – mēneša dienu, jo gadā nav 24 mēnešu, bet gada skaitlis jau ir zināms, un tāpēc 12 apzīmē atlikušo – mēnesi. Ja mēneša diena vai mēneša kārtas numurs ir viencipara skaitlis, tad to papildina līdz diviem cipariem, sākumā pierakstot nulli. Tāpēc 02/03/04 var apzīmēt 6 dažādus datumus, jo katrs no skaitļiem der gan kā gada, gan mēneša, gan kā mēneša dienas apzīmējums jebkurā iespējamajā kombinācijā.

Uzrakstiet programmu, kas dotam datuma pierakstam formā xx/yy/zz nosaka, cik dažādus datumus šādi pierakstītais datums īstenībā varētu apzīmēt.

Piezīmes: Vadīties pēc Gregora kalendāra. Ar XXI gadsimtu tiek saprasti visi gadi no 2000. līdz 2099. gadam. XXI gadsimtā visi gadi, kuru kārtas skaitlis dalās ar 4, ir garie gadi.

Ievaddati

Teksta faila `datumi.in` vienīgajā rindā atrodas datuma apzīmējums formā `xx/yy/zz` (divi cipari, simbols “/”, divi cipari, simbols “/”,divi cipari).

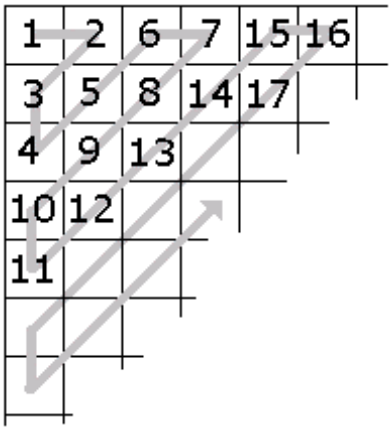
Izvaddati

Teksta failam `datumi.out` vienīgajā rindā jāizvada viens vesels nenegatīvs skaitlis - dažādo iespējamo pareiza datuma interpretāciju skaits.

Piemēri

Ievaddati (fails `datumi.in`)	Izvaddati (fails `datumi.out`)
`86/12/24`	`1`
`02/03/04`	`6`
`01/01/01`	`1`
`97/98/99`	`0`
`04/04/29`	`2`
`02/02/29`	`1`

5. ”ŠAHA ZIRDZIŅA CEĻOJUMS”

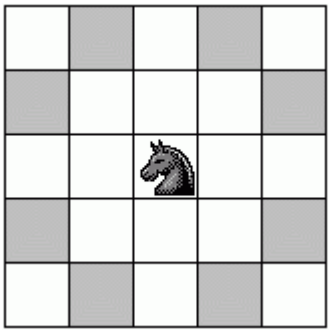


1.zīm. Rūtiņu numerācija

Ļoti, ļoti lielas rūtiņu lapas visas rūtiņas pēc kārtas sanumurētas tā, kā redzams 1.zīmējumā.

Kādā no laukuma rūtiņām novietots šaha zirdziņš. Katrā gājienā zirdziņš pēc šaha noteikumiem (2.zīmējumā pelēkā krāsā iekrāsotas visas tās rūtiņas, uz kurām zirdziņš drīkst izdarīt nākošo gājienu) pārvietojas uz to iepriekš neapmeklēto rūtiņu, kuras numurs ir mazākais iespējamais. Zirdziņš nevienā brīdī nedrīkst "nokāpt" no lapas. Piemēram, ja sākumā zirdziņš atrodas 6.rūtiņā, tad nākošajā gājienā zirdziņš pārvietosies uz 3.rūtiņu (jo pārējo trīs atļauto gājienu rūtiņās ierakstītie skaitļi ir lielāki - 9, 17 un 18).

Uzrakstiet programmu, kas ievadītam zirdziņa sākotnējās rūtiņas numuram N un gājienu skaitam K nosaka, kāds būs tās rūtiņas numurs, kurā pēc K-tā gājiena izdarīšanas atradīsies zirdziņš!


2.zīm. Zirdziņa iespējamie gājieni

Ievaddati

Teksta faila `zirgs.dat` vienīgajā rindā dotas divu naturālu skaitļu N (1≤N≤100000) un K (1≤K≤100000) vērtības, kas atdalītas ar tukšumsimbolu.

Izvaddati

Teksta faila `zirgs.rez` vienīgajā rindā jāizvada naturāls skaitlis - tās rūtiņas numurs, kurā zirdziņš atradīsies pēc K-tā gājiena izdarīšanas.

Ja, izdarot gājienus, kādā brīdī iestājas situācija, ka zirdziņam nav kur iet (visās atļautajās rūtiņās zirdziņš jau ir bijis iepriekš), tad failā jāizvada vārds "NEVAR", kam seko tās rūtiņas numurs, no kuras zirdziņš nevarēja izdarīt gājienu. Starp "NEVAR" un rūtiņas numuru jābūt tieši vienam tukšumsimbolam.

Piemēri

Ievaddati (fails `zirgs.dat`)	Izvaddati (fails `zirgs.rez`)
`5 1`	`7`
`1 4`	`13`
`13 13`	`16`
`100 4000`	`NEVAR 45`

6."STUDENTU BALLĪTE"

Kā jau katru gadu pēc nokārtotas un/vai pagājušas sesijas studenti gribēja nosvinēt šo priecīgo pasākumu ar kāda augstvērtīga dzēriena baudīšanu filozofiskās noskaņās. Katrs no svinību dalībniekiem šim cēlajam mērķim ziedoja noteiktu naudas daudzumu tā, ka beigās bija savākti N santīmi. Pēc tam visi ziedotāji sapulcējās un nolēma, ka augstvērtīgā dzēriena lomu šajā svinīgajā sanākšanā pildīs Piebalgas Dzēriens (PD). Par laimi starp sapulces dalībniekiem bija gan matemātikas, gan ekonomikas zinātnēs izglītoti cilvēki, kuri nosprieda, ka nepieciešams pētījums – cik maksā viena PD pudele (PDp) un par cik var pārdot vienu tukšu PDp. Pēc kāda laika tika noskaidrots, ka viena pilna PDp maksā X santīmus, bet par tukšu PDp taras punktā var saņemt Y santīmus.

Paredzamā svētku dienas kārtība bija šāda:

- 11.00-12.00 Pulcēšanās,
- 12.00-12.30 Svinību vietas sakopšana un iztukšoto PDp savākšana,
- 12.30-14.00 Tukšo PDp nodošana taras punktā (pudeļu apmaiņa pret santīmiem),
- 14.00-14.15 Maksimāla daudzuma pilnu PDp iegāde (pudeļu skaits vienmēr ir naturāls skaitlis un naudas atlikums (ja tāds ir), paliek uz nākošo dienu),
- 14.15-22.00 PD baudīšana filozofiskās noskaņās.

Kā redzams pēc dienas kārtības, tad svētku kopējais dienu skaits nav norādīts – tas ir tāpēc, ka neviens no svinību dalībniekiem nevēlējās uzņemties veikt aprēķinu, cik dienas ilgs svinības, ja pirmajā dienā nav nevienas tukšas PDp, bet ir N santīmi, un svinības tiek rīkotas tikmēr, kamēr plkst. 14.16 ir kaut viena pilna PDp. Vēl jāņem vērā, ka vienīgā iespēja iegūt naudu ir nododot tukšās PDp.

Ievaddati

Teksta failā party.in pirmajā rindiņā ir ierakstīts naturāls skaitlis T(T≤5), kas norāda, cik ballīšu apraksti sekos. Katrā no nākamajām T rindiņām ir dots vienas ballītes apraksts, kas sastāv no trim veseliem skaitļiem N (1≤N≤1000000), X(2≤X≤1000000) un Y(1≤Y<X), kas atdalīti ar tukšuma simbolu. N norāda, cik santīmus studenti ir saziedojuši plānojot ballīti, X – cik maksā viena pilna PDp , bet Y norāda, par cik santīmiem ir iespējams nodot tukšu PDp.

Izvaddati

Teksta failā party.out jābūt tieši T rindiņām, katrā pa vienam veselam skaitlim tā, lai izvaddatu faila i-tajā rindiņā būtu rakstīts, cik dienas studenti varēs rīkot ballīti, kas aprakstīta ievaddatu faila i+1-jā rindiņā.

Piemērs

Ievaddati (fails party.in)	Izvaddati (fails party.out)
3 8 4 2 7 4 2 5 4 2	2 2 1

7."KANTORA SKAITĻI"

Ciparu virkni, kas sastāv tikai no nullēm un vieniniekiem, veido pa soļiem sekojoši.

Sākumā (0-tajā solī) virkne sastāv no viena vieninieka: "1".

Katrā nākošajā solī virkni veido no iepriekšējā solī iegūtās virknes, katru nulli tajā aizvietojot ar "000", bet katru vieninieku – ar "101".

Pirmajā solī tiks izveidota virkne "101", otrajā solī – "101000101", trešajā – "10100010100000000101000101", utt.

Uzrakstiet programmu, kas nosaka, kurā N-tajā solī izveidotās virknes pozīcijā pēc kārtas (pozīciju numerācija sākas no virknes kreisā gala ar 1) atrodas K-tais vieninieks!

Ievaddati

Teksta faila `kantors.in` pirmajā rindā ir dots vesels nenegatīvs skaitlis N($0 \le N \le 500$), otrajā rindā – naturāls skaitlis K. Ir zināms, ka N-tajā solī izveidotajā virknē vieninieku skaits ir vismaz K.

Izvaddati

Teksta faila `kantors.out` vienīgajā rindā jāizvada naturāls skaitlis – pozīcija virknē, kurā atrodas pēc kārtas K-tais vieninieks.

Piemēri

Ievaddati (fails `kantors.in`)	Izvaddati (fails `kantors.out`)
2 3	7
3 6	21
45 10000000000	11580743424275073

Piezīme

Uzdevums nosaukts par godu Georgam Ferdinandam Ludvigam Filipam Kantoram (1845-1918), un uzdevuma ideja ir radusies no *Kantora kopām*.