

LATVIJAS REPUBLIKAS 14. INFORMĀTIKAS OLIMPIĀDES IETEICAMIE I POSMA UZDEVUMI

1. "SUMMA"

Uz papīra lapas tika uzrakstīts naturāls skaitlis $\overline{a_n a_{n-1} \dots a_2 a_1}$, kura visi cipari bija lielāki par 0. Skaitlim bija ne vairāk kā 100 cipari. Pēc tam zem šī skaitļa stabiņā tika uzrakstīti skaitļi $\overline{a_{n-1} \dots a_2 a_1}$, $\overline{a_{n-2} \dots a_2 a_1}$, ..., $\overline{a_2 a_1}$, $\overline{a_1}$. Beigās visi skaitļi tika saskaitīti. Piemērs, kad sākotnējais skaitlis ir 7231493, parādīts zīmējumā.

```
7231493
 231493
  31493
   1493
    493
     93
      3
-----
7496561
```

Uzrakstiet programmu, kas ievadītam sākotnējam skaitlim izvada aprēķināto summu.

Ievaddati

Teksta faila SUMMA.DAT vienīgajā rindā dots sākumā uz lapas uzrakstītais skaitlis.

Izvaddati

Teksta faila SUMMA.REZ vienīgajā rindā jāizvada viens skaitlis - aprēķinātā summa.

Piemērs

Ievaddati(fails SUMMA.DAT)
7231493

Izvaddati(fails SUMMA.REZ)
7496561

2."CEĻOJUMS"

Bezgalīgas kvadrātveida rūtiņu lapas kādas rūtiņas virsotnē atrodas rūķītis. Rūķītis māk pārvietoties tikai pa rūtiņu malām. Vienas rūtiņas malas garums ir viens rūķīša solis. Katrā brīdī rūķīša skats ir pavērsts pret kādu no blakus esošajām rūtiņu virsotnēm.

Bieži rūķītis devās ceļojumā, pa ceļam pierakstot instrukcijas, kā viņš ir gājis, lai šo ceļojumu būtu iespējams atkārtot. Instrukcija var būt:

- **K** (nozīmē, ka šajā virsotnē rūķītis pagriežies 90° pa kreisi;
- **L** (nozīmē, ka šajā virsotnē rūķītis pagriežies 90° pa labi;
- **S** <soļu skaits> (nozīmē, ka izvēlētajā virzienā rūķītis nogājis *soļu skaits* soļus. Starp burtu un skaitli pierakstā ir viens tukšumsimbols);
- **V** (apzīmē ceļojuma beigas – vienmēr tikai pēdējā instrukcija)

Uzrakstiet programmu, kas dotam ceļojuma pierakstam nosaka mazāko soļu skaitu, kāds rūķītim nepieciešams, lai atgrieztos sākumpunktā (tas var būt mazāks par ceļojuma kopgarumu)!

Ievaddati

Teksta faila TAKA.DAT katrā rindā dota viena instrukcija. Kopējais instrukciju skaits nepārsniedz 10000. Vienā reizē veikto soļu skaits (instrukcija S) nepārsniedz 1000.

Izvaddati

Teksta faila TAKA.REZ vienīgajā rindā jāizvada vesels skaitlis – mazākais soļu skaits, kāds nepieciešams, lai no ceļojuma beigu punkta atgrieztos sākumpunktā.

Piemērs

Ievaddati(fails TAKA.DAT)

S 7
L
S 1
K
S 4
K
S 3
K
S 8
V

Izvaddati(fails TAKA.REZ)

5

3."IEKAVAS"

Korektu iekavu izteiksmi definē šādi:

- () ir korekta iekavu izteiksme;
- ja A ir korekta iekavu izteiksme, tad (A) arī ir korekta iekavu izteiksme;
- ja A un B ir korektas iekavu izteiksmes, tad AB arī ir korekta iekavu izteiksme.

Tā, piemēram, $((()))$ un $()()((()))$ ir korektas iekavu izteiksmes, bet $)()$ (un $()$) –nav.

Korektā iekavu izteiksmē nodzēsa dažas iekavas.

Uzrakstiet programmu, kas nosaka mazāko iespējamo nodzēsto iekavu skaitu!

Ievaddati

Teksta faila IEKAVAS.DAT pirmajā rindā dota simbolu virkne, kas sastāv tikai no atverošajām un aizverošajām iekavām. Virknes garums ir vismaz viens un ne vairāk kā 100 simboli.

Izvaddati

Teksta faila IEKAVAS.REZ vienīgajā rindā jāizvada vesels skaitlis - mazākais iekavu skaits, kādu korektā iekavu virknē bija jānodzēš, lai iegūtu doto.

Piemēri

Ievaddati(fails IEKAVAS.DAT)
) (() ()) (

Izvaddati(fails IEKAVAS.REZ)
3

Ievaddati(fails IEKAVAS.DAT)
(() ())

Izvaddati(fails IEKAVAS.REZ)
0

4."ŠAHS"

Tiek ievadīta šaha laukuma lauciņu koordināšu virkne (virknes garums nepārsniedz 100). Noskaidrot, vai šī virkne atbilst šaha zirdziņa (Z), laidņa (L), dāmas (D), torņa (T) vai karaļa (K) pēc kārtas izdarītu gājienu virknei! Virknes pirmais lauciņš ir tas šaha laukuma lauciņš, kur figūra atrodas ceļojuma sākumā (tas drīkst būt jebkurš šaha laukuma lauciņš). Pēc katra gājiena figūrai jāpārvietojas uz citu lauciņu (tā nevar stāvēt uz vietas). Figūra sava ceļojuma laikā vienā un tajā pašā rūtiņā drīkst nonākt vairākkārt.

Ievaddati.

Teksta faila IEVADS.TXT pirmajā rindā dots naturāls skaitlis n - virknē uzrādīto lauciņu skaits. Sekojošās n faila rindās dotas rūtiņu koordinātas (viena koordināta katrā rindā, katra koordināta formā latīņu alfabēta mazais burts (no "a" līdz "h") un cipars (no "1" līdz "8")).

Izvaddati.

Teksta faila IZVADS.TXT vienīgajā rindā jāizvada burtu virkne, kas sastāv no viena vai vairākiem burtiem.

Ja dotā rūtiņu virkne atbilst kādas figūras sekojošu gājienu virknei, tad jāizvada šai figūrai atbilstošais burts. Ir iespējams, ka viena virkne atbilst vairākām figūrām un tādā gadījumā jāizvada visu derīgo figūru burti (katrs burts vienu reizi, burtu secība var būt patvaļīga).

Ja šādu figūru atrast nav iespējams, jāizvada vārds "NEDER".

1.piemērs

Ievaddati(fails IEVADS . TXT)
3
a1
c3
e5

Izvaddati(fails IZVADS . TXT)
LD

2.piemērs

Ievaddati(fails IEVADS . TXT)
4
a1
c2
a3
b4

Izvaddati(fails IZVADS . TXT)
NEDER

5."ADRESES"

INTERNET protokola (IP) adrese sastāv no četriem naturāliem skaitļiem, katrs no kuriem nepārsniedz 999. Skaitļi tiek atdalīti ar punktiem.

Tā, piemēram, 122.10.999.1 un 1.23.45.678 ir IP adreses, bet 7.07.1.1, 1.2.3.4.5, 11.12.0.13 un 2.34.56.7890 – nav.

IP adresē nodzēsa visus punktus un ieguva vienu garu skaitli. Uzrakstiet programmu, kas ievadītam skaitlim nosaka, cik ir tādas IP adreses, no kurām varēja iegūt šo skaitli!

Ievaddati

Teksta faila ADRESES.DAT vienīgajā rindā dots naturāls skaitlis, kura ciparu skaits nepārsniedz 12.

Izvaddati

Teksta faila ADRESES.REZ vienīgajā rindā jāizvada vesels skaitlis – to IP adresu skaits, no kurām iepriekšminētajā veidā varēja iegūt doto skaitli.

Piemērs

Ievaddati(fails ADRESES.DAT)
1503175417

Izvaddati(fails ADRESES.REZ)
7

Piezīme: šīs adreses ir

1.503.175.417

150.3.175.417

150.31.75.417

150.31.754.17

150.317.5.417

150.317.54.17

150.317.541.7

6."SALAS"

Kādas valsts iedzīvotāji dzīvo uz n salām. Salas ir sanumurētas ar naturāliem skaitļiem no 1 līdz n . Starp dažām no tām ir nodibināta kuģīšu satiksme. Satiksme ir organizēta tā, ka no katras salas ir iespējams tikt uz katru citu izmantojot tikai kuģīšu satiksmi (ja starp attiecīgajām salām nav tiešās satiksmes, tad pārsēžoties). Kuģīšu dažādo tipu dēļ var izrādīties, ka ne vienmēr tiešā satiksme ir arī tā ātrākā.

Tā, piemēram, ja ir 3 salas un brauciens no pirmās uz otro salu aizņem 30 minūtes, no otrās uz trešo – 15 minūtes, bet no pirmās uz trešo – 50 minūtes, tad no pirmās uz trešo salu visātrāk iespējams nokļūt 45 minūtēs (jābrauc vispirms uz otro salu). Zināms, ka katrs kuģītis kursē tikai starp divām noteiktām salām. Starp katrām divām salām kursē ne vairāk kā viens kuģītis.

Uzrakstiet programmu, kas nosaka, kāds mazākais laiks nepieciešams, lai no katras salas nokļūtu uz jebkuru citu! Laiku, kas jāpavada, gaidot kuģīti, nav jāņem vērā.

Ievaddati

Teksta faila SALAS.DAT pirmajā rindā doti divi naturāli skaitļi – salu skaits n ($n \leq 50$) un kuģīšu maršrutu skaits k ($k \leq n(n-1)/2$).

Nākošajās k faila rindās doti kuģīšu maršrutu apraksti - pa vienam maršruta aprakstam katrā rindā. Katra maršruta apraksts ir formā $\langle x \rangle \langle y \rangle \langle \text{brauciena laiks minūtēs} \rangle$, kur x un y – to salu numuri, starp kurām kursē dotā maršruta kuģītis. Nevienam maršrutam brauciena laiks nepārsniedz 100 minūtes. Starp katriem diviem blakus skaitļiem atrodas viens tukšumsimbols. Katra maršruta apraksts failā dots tieši vienreiz.

Izvaddati

Teksta failam SALAS.REZ jāsatur tieši n rindas. Katrā rindā jāizvada n veseli skaitļi, kas atdalīti ar tukšumsimboliem. i -tajā rindā j -tais izvadītais skaitlis atbilst mazākajam laikam minūtēs, kāds nepieciešams, lai no i -tās salas nokļūtu līdz j -tajai salai.

Piemērs

Ievaddati (fails SALAS.DAT)

```
3 3
2 3 15
1 3 50
2 1 30
```

Izvaddati (fails SALAS.REZ)

```
0 30 45
30 0 15
45 15 0
```