

LATVIJAS 16. INFORMĀTIKAS OLIMPIĀDES III POSMA UZDEVUMU APSKATS
Pirmā diena (2003.gada 24.marts)



Uzdevuma nosaukums:	Punkti	Konferences	Mazdārziņi
Programmas nosaukums:	punkti.exe	konfer.exe	darzini.exe
Ievaddatu faila nosaukums:	punkti.dat	konfer.dat	darzini.dat
Izvaddatu faila nosaukums:	punkti.rez	konfer.rez	darzini.rez
Izpildes laika ierobežojums vienam testam* :	5 sekundes	1 sekunde	1 sekunde
Testu skaits:	13	12	10
Piezīmes par vērtēšanu:			
Maksimāli iespējamais punktu skaits par uzdevumu:	100	100	100
Palīgfaili uz dalībnieka disketes	Ievaddatu faila piemērs Izvaddatu faila piemērs	Ievaddatu faila piemērs Izvaddatu faila piemērs	Ievaddatu faila piemērs Izvaddatu faila piemērs Izvaddatu pārbaudes programma darztest.exe

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili atrodas tekošajā katalogā)!

Neaizmirstiet saglabāt savas programmas (izejas tekstus un .exe moduļus) ar norādītajiem nosaukumiem uz disketēm saknes katalogā!

* Izpildes laika ierobežojums norādīts datoram, uz kura programmas tiks testētas. Sacensību un testa datora ātrdarbības salīdzināšanai izmantojiet programmu « **atrums.exe** », kas atrodas uz disketes.

Jautājumus par uzdevumu formulējumiem centieties uzdot sacensību pirmās stundas laikā!

LATVIJAS 16. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Pirmā diena (2003.gada 24.marts)



1. "PUNKTI"

Plaknē doti n dažādi punkti, kuru koordinātas ir veseli skaitļi.

Uzrakstiet programmu, kas nosaka, kāds lielākais punktu skaits atrodas uz vienas taisnes!

Ievaddati

Teksta faila `punkti.dat` pirmajā rindā dots naturāls skaitlis n - punktu skaits ($1 < n \leq 1000$). Katrā no nākošajām n rindām doti divi veseli skaitļi, kas atdalīti ar tukšumsimbolu - kārtējā punkta x un y koordinātas. Neviena skaitļa vērtība pēc moduļa nepārsniedz 30000.

Izvaddati

Teksta faila `punkti.rez` vienīgajā rindā jāizvada lielākais punktu skaits, kas atrodas uz vienas taisnes.

Piemērs

Ievaddati (fails <code>punkti.dat</code>)	Izvaddati (fails <code>punkti.rez</code>)
5	3
1 1	
0 7	
2 2	
-2003 -2003	
5000 -4444	

2. "KONFERENCES"

Zinātnieks šodien ir saņēmis konferenču grafiku tuvākajiem gadiem. Katrai konferencē ir zināms tās sākuma un beigu datums. Zinātnieks vēlas apmeklēt pēc iespējas vairāk konferenču. Tā kā zinātnieks ir apzinīgs, viņš tajās konferencēs, kuras apmeklē, piedalās no pirmās līdz pēdējai dienai bez pārtraukumiem. Pārbraukšanai uz citu konferenci pilnībā pietiek ar nakti - ja vienā dienā kāda konference beidzas, tad jau nākamajā dienā zinātnieks var būt uz citas konferences sākumu.

Dotam konferenču grafikam nosakiet, kādu lielāko skaitu konferenču zinātnieks var apmeklēt!

Ievaddati

Faila `konfer.dat` pirmajā rindā dots konferenču skaits n kā naturāls skaitlis, kura vērtība nepārsniedz 1000. Katrā no nākošajām n faila rindām dota informācija par vienu konferenci - tās sākuma un beigu datumi, kas atdalīti ar tukšumsimbolu. Katrs no datumiem ir formā `dd.mm.yyyy`, kur `dd` apzīmē dienu (divi cipari, robežās no 01 līdz maksimālam dienas numuram attiecīgajā mēnesī), `mm` - mēnesi (divi cipari, robežās no 01 līdz 12), bet `yyyy` - gadu (četri cipari, robežās no 2004 līdz 2099).

Konferences beigu datums nevar būt agrāk par sākuma datumu (datumi var sakrist, kas nozīmē, ka konference notiek vienu dienu). Konferences ievaddatos var nebūt sakārtotas hronoloģiskā secībā.

Izvaddati

Faila `konfer.rez` vienīgajā rindā jāizvada viens naturāls skaitlis - maksimālais konferenču skaits, ko zinātnieks var apmeklēt.

Piemērs

Ievaddati(fails <code>konfer.dat</code>)	Izvaddati(fails <code>konfer.rez</code>)
5	3
01.01.2004 01.03.2004	
28.02.2004 28.02.2004	
31.01.2004 04.02.2004	
01.03.2004 24.06.2004	
22.02.2004 21.02.2005	

3. "MAZDĀRZIŅI"

Tuntuļu pagasta dome mazdārziņu izveidošanai ir atvēlējusi $N \times N$ rūtiņu lielu zemes laukumu, kas sadalīts starp M īpašniekiem. Katrs no īpašniekiem ir iegādājies noteiktu daudzumu rūtiņu un katram īpašniekam tā identifikācijai ir piešķirts latīņu alfabēta lielais vai mazais burts (lielie un mazie burti šī uzdevuma izpratnē atšķiras). Katram īpašniekam var piederēt tikai viens zemes gabals. Pārdoto rūtiņu kopskaits ir N^2 . Diemžēl vēl pirms pilnīgas dokumentu nokārtošanas un robežu precīzas nospraūšanas visi īpašnieki sākuši dārza mājiņu celtniecību, pārējās rūtiņas atstājot tukšas.

A			
	c		X
		a	

1. zīm. Dārza mājiņu izvietojums

Schematisks mazdārziņu plāna piemērs redzams 1. zīmējumā, kur katru īpašnieku identificējošais burts ir ierakstīts tajā rūtiņā, kur sāka dārza mājiņas celtniecību.

Lai pabeigtu dokumentu kārtošanu, nepieciešams noteikt, kura rūtiņa kuram īpašniekam pieder. Pie kam papildus jāievēro sekojoši noteikumi:

- rūtiņa, kurā zemes īpašnieks sācis celt dārza mājiņu, pieder šim īpašniekam
- katra zemes īpašniekam piederošā rūtiņa atrodas vienā rindā vai kolonnā ar to rūtiņu, kurā atrodas viņam piederošā dārza mājiņa
- katrs īpašnieks no savas dārza mājiņas var aiziet līdz jebkurai sev piederošai rūtiņai visu laiku ejot pa sev piederošām rūtiņām, pārejot pār rūtiņu kopīgo malu
- katram īpašniekam jāierāda tieši tik rūtiņas, cik viņš ir iegādājies.

Ja iepriekš aplūkotajā piemērā **A** būtu iegādājies piecas, **c** - divas, **a** - piecas, bet **X** - četras rūtiņas, tad rūtiņu piederība varētu būt tāda, kā parādīts 2. zīmējumā. Šajā piemērā rūtiņu piederība varēja būt arī citāda.

Uzrakstiet programmu, kas dotam dārza mājiņu novietojumam un nopirkto rūtiņu skaitam nosaka, kura rūtiņa kuram īpašniekam pieder!

1	A	A	A	X
2	A	c	X	X
3	A	c	a	X
4	a	a	a	a
	1	2	3	4

2. zīm. Rūtiņu piederība

Ievaddati

Faila `darzini.dat` pirmajā rindā doti divi naturāli skaitļi N (mazdārziņiem atvēlētā zemes gabala rūtiņu rindu un kolonnu skaits, $N \leq 30$) un M (zemes īpašnieku skaits, $M \leq \min(N^2, 52)$), kas atdalīti ar tukšumsimbolu. Katrā no nākošajām M faila rindām dota informācija par vienu īpašnieku sekojošā formā:

`<īpašnieka_id> <nopirkto_rūtiņu_skaits> <rindas_nr> <kolonnas_nr>`

`<īpašnieka_id>` ir viens latīņu alfabēta burts - īpašnieka identifikators,

`<nopirkto_rūtiņu_skaits>` ir naturāls skaitlis, kas apzīmē attiecīgā īpašnieka nopirkto rūtiņu skaitu,

`<rindas_nr>` ir tās rindas numurs, kur atrodas šim īpašniekam piederošā dārza mājiņa (rindas tiek numurētas Ziemeļu-Dienvidu virzienā pēc kārtas no 1 līdz N),

`<kolonnas_nr>` ir tās kolonnas numurs, kur atrodas šim īpašniekam piederošā dārza mājiņa (kolonnas tiek numurētas Rietumu-Austrumu virzienā pēc kārtas no 1 līdz N).

Pēc identifikatora kā arī starp skaitļiem ievaddatos ir pa vienam tukšumsimbolam.

Zināms, ka katram no dotajiem ievaddatiem būs iespējams atrast korektu atrisinājumu.

Ja iespējami vairāki atrisinājumi, izvadiet vienu no tiem.

Izvaddati

Failam `darzini rez` jāsaturs tieši N rindas. Katrā rindā jāizvada tieši N simboli (bez atdalošiem tukšumsimboliem). Faila i -tās rindas j -tajam simbolam jābūt tā īpašnieka identifikatoram, kam pieder zemesgabala i -tās rindas j -tā rūtiņa.

Piemērs

Ievaddati(fails `darzini.dat`)

```
4 4
A 5 1 1
c 2 2 2
X 4 2 4
a 5 4 3
```

Izvaddati(fails `darzini rez`)

```
AAAX
AcaX
AcaX
aaaX
```

LATVIJAS 16. INFORMĀTIKAS OLIMPIĀDES III POSMA UZDEVUMU APSKATS
Otrā diena (2003.gada 25.marts)



Uzdevuma nosaukums:	Šablons	Pārmijnieks
Programmas nosaukums:	sablons.exe	parmijas.exe
Ievaddatu faila(u) nosaukums(i):	labie.dat sliktie.dat	parmijas.dat
Izvaddatu faila nosaukums:	sablons.rez	parmijas.rez
Izpildes laika ierobežojums vienam testam*:	2 sekundes	1 sekunde
Testu skaits:	12	13
Piezīmes par vērtēšanu:		
Maksimāli iespējamais punktu skaits par uzdevumu:	100	100
Palīgfaili uz dalībnieka disketes	Ievaddatu failu piemērs Izvaddatu faila piemērs	Ievaddatu faila piemērs Izvaddatu faila piemērs

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili atrodas tekošajā katalogā)!

Neaizmirstiet saglabāt savas programmas (izejas tekstus un .exe moduļus) ar norādītajiem nosaukumiem uz disketēm saknes katalogā!

* Izpildes laika ierobežojums norādīts datoram, uz kura programmas tiks testētas. Sacensību un testa datora ātrdarbības salīdzināšanai izmantojiet programmu « **atrums.exe** », kas atrodas uz disketes.

Jautājumus par uzdevumu formulējumiem centieties uzdot sacensību pirmās stundas laikā!

LATVIJAS 16. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Otrā diena (2003.gada 25.marts)



1. "ŠABLONS"

Lai ātrāk atrastu vairākas interesējošas simbolu virknes (piemēram, failu nosaukumus), datoru vidē lieto simbolu virkņu *šablonus*. Šablons ir simbolu virkne, kas var saturēt tikai latīņu alfabēta mazos burtus, ciparus, jautājuma zīmes un zvaigznītes. Šablonā divas vai vairāk zvaigznītes nedrīkst atrasties blakus. Derīgi šabloni ir, piemēram "ab*", "ne?z*inu", "??3", "*1*", bet nav "A?", "#13", "a**b".

Teiksim, ka simbolu virkne atbilst šablonam, ja katram šablona burtam virknē atbilst tieši šis pats burts (lielais vai mazais), katram ciparam atbilst tieši šis cipars, jautājuma zīmei atbilst jebkurš viens burts vai cipars, bet zvaigznītei - jebkura pēc kārtas ņemtu simbolu virkne (tai skaitā tukšā virkne).

Tā, piemēram, šablonam "?a*n?" atbilst virknes "gans", "salons", "PAUNA", bet neatbilst "cirvis", "spainis" un "SVINS".

Kazimirs ir iedomājies kādu šablonu un izveidojis divus vārdu sarakstus - "labo" vārdu sarakstu, kurā esošie vārdi atbilst iedomātajam šablonam un "slikto" vārdu sarakstu, kurā esošie vārdi iedomātajam šablonam neatbilst.

Uzrakstiet programmu, kas pēc šiem diviem vārdu sarakstiem atrod Kazimira iedomāto šablonu!

Ievaddati

Teksta faila `labie.dat` pirmajā rindā dots naturāls skaitlis N_{LABIE} - to do to vārdu, kas atbilst Kazimira iedomātajam šablonam, skaits. Katrā no nākošajām N_{LABIE} rindām dots pa vienam vārdam, kas atbilst šablonam.

Teksta faila `sliktie.dat` pirmajā rindā dots naturāls skaitlis N_{SLIKTIE} - to do to vārdu, kas neatbilst Kazimira iedomātajam šablonam, skaits. Katrā no nākošajām N_{SLIKTIE} rindām dots pa vienam vārdam, kas neatbilst šablonam.

Katrs no ievaddatu failos dotajiem vārdiem var saturēt tikai latīņu alfabēta lielos un mazos burtus un ciparus. Neviena dotā vārda garums nepārsniedz 15 simbolus. Katrā failā būs dots vismaz viens un ne vairāk kā desmit vārdi.

Izvaddati

Teksta faila `sablons.rez` vienīgajā rindā jāizvada iespējamais Kazimira iedomātais šablons. Ja iespējami vairāki atrisinājumi, jāizvada jebkurš no tiem. Zināms, ka katram testam, kas tiks izmantots programmas pārbaudei, vismaz viens šāds šablons eksistē.

Piemērs

Ievaddati (fails `labie.dat`)

3

lausks

tase3

TaNtUkS

Ievaddati (fails `sliktie.dat`)

2

saule

aukstums

Izvaddati (fails `sablons.rez`)

?a*s*

2. "PĀRMIJNIEKS"

Pilsētas tramvaju depo atrodas ārpus pilsētas un no tā uz pilsētu ved viens sliežu ceļš. Katru rītu noteikts tramvaju skaits no depo izbrauc uz līnijas, bet vakarā atgriežas tajā. Pavisam depo ir 2^n tramvaji, kas sanumurēti ar naturāliem skaitļiem no 1 līdz 2^n pēc kārtas. Katram tramvajam depo ēkā ir noteikta vieta - tramvajs ar mazāku numuru atrodas ēkā vairāk pa kreisi. Depo shēma pie $n=3$ ir redzama 1.zīmējumā.

Ar taisnes nogriežņiem ir attēloti sliežu posmi, bet katram no shēmā redzamajiem aplīšiem atbilst vienkārša pārmija, kas katru brīdi var būt ieslēgta vienā no diviem stāvokļiem - "pa kreisi" vai "pa labi" kā redzams 2.zīmējumā.

Pārmijas depo teritorijā ir izkārtotas n līmeņos - pirmajā līmenī ir viena pārmija, otrajā līmenī divas, n -tajā līmenī 2^{n-1} pārmija. Katra i -tā līmeņa pārmija ($1 < i < n$) savieno vienu $i-1$ -ā līmeņa pārmiju ar divām $i+1$ -ā līmeņa pārmijām. Katra n -tā līmeņa pārmija savieno kādu $n-1$ -ā līmeņa pārmiju ar diviem depo nodalījumiem, bet pirmā līmeņa pārmija savieno divas otrā līmeņa pārmijas ar ceļu uz pilsētu.

Lai tramvajs varētu izbraukt uz līnijas, visām pārmijām no depo attiecīgā nodalījuma līdz ceļam uz pilsētu jābūt pārslēgtajām pareizajos stāvokļos. Tā piemēram, lai uz līnijas varētu izbraukt 4.tramvajs, attiecīgajām otrā un trešā līmeņa pārmijām jābūt pārslēgtām pa labi, bet pirmā līmeņa pārmijai - pa kreisi kā redzams 3.zīmējumā.

Katru rītu pirms tramvaju izlaišanas uz līnijas visas pārmijas atrodas stāvoklī "pa kreisi".

Pārmiju korekta pārslēgšana ir pārmijnieka Jāņa ziņā. Jānim tiek paziņots, kuri tramvaji no rīta ir jāizlaiž uz līnijas un Jānim ir jāizvēlas to izbraukšanas secība un jānodrošina pārmiju pareiza pārslēgšana. Jānis ir ievērojis, ka kopējais pārmiju pārslēgšanu skaits var būt atkarīgs no tā, kādā secībā tramvaji tiek izlaisti uz līnijas.

Tā, piemēram, ja uz līnijas ir jāizlaiž tikai 4. un 5. tramvajs, tad, izlaižot tos tieši šādā secībā, būs nepieciešamas 3 pārslēgšanas, bet pretējā (vispirms 5., tad 4.) - četras pārslēgšanas.

Jānis ir sapratis, ka nepieciešams izstrādāt datorprogrammu, kas patvaļīgai n vērtībai un dotiem tramvaju numuriem noteiktu mazāko pārmiju pārslēgšanu skaitu, kas jāveic, lai visi norādītie tramvaji tiktu izlaisti uz līnijas.

Uzrakstiet šādu datorprogrammu!

Ievaddati

Teksta faila `parmijas.dat` pirmajā rindā doti divi naturāli skaitļi n ($1 \leq n \leq 30$) un m ($1 \leq m \leq \min(2^n, 3000)$), kas atdalīti ar tukšumsimbolu. n ir pārmiju līmeņu skaits, bet m - uz līnijas izlaižamo tramvaju skaits. Katrā no nākošajām m faila rindām dots pa vienam naturālam skaitlim - tramvaja numuram, kas jāizlaiž uz līnijas. Visi dotie tramvaju numuri savā starpā ir atšķirīgi.

Izvaddati

Teksta faila `parmijas.rez` vienīgajā rindā jāizvada viens vesels nenegatīvs skaitlis - mazākais nepieciešamais pārmiju pārslēgšanu skaits.

Piemērs

Ievaddati (fails `parmijas.dat`)

3 2

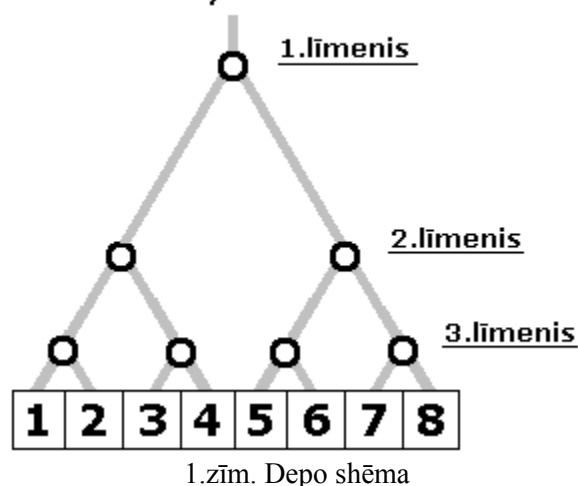
5

4

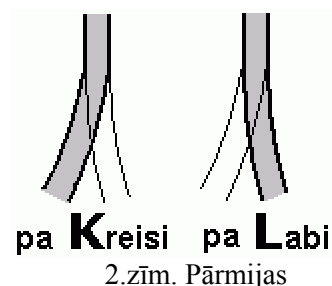
Ievaddati (fails `parmijas.rez`)

3

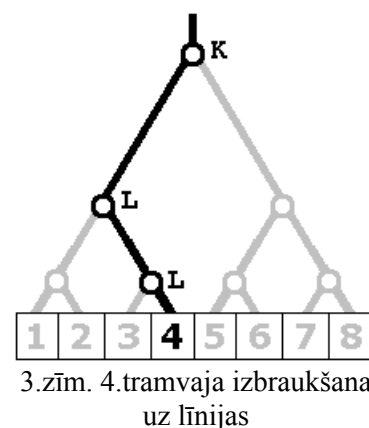
Uz pilsētu



1.zīm. Depo shēma



2.zīm. Pārmijas



3.zīm. 4.tramvaja izbraukšana uz līnijas