



Uzdevuma nosaukums:	Datums	Bobslejs	Nederīgie autobusi
Ievaddatu faila nosaukums:	datums.dat	bobslejs.dat	autobusi.dat
Izvaddatu faila nosaukums:	datums.rez	bobslejs.rez	autobusi.rez
Klases vārds risinājumam valodā <i>Java:</i>	Datums	Bobslejs	Autobusi
Izpildes laika ierobežojums vienam testpiemēram sekundēs (laiks tiek mērīts uz testēšanas servera):	<i>Pascal: 0,1</i> <i>C: 0,1</i> <i>C++: 0,1</i> <i>Java: 0,3</i>	<i>Pascal: 0,2</i> <i>C: 0,2</i> <i>C++: 0,2</i> <i>Java: 0,5</i>	<i>Pascal: 0,2</i> <i>C: 0,2</i> <i>C++: 0,2</i> <i>Java: 0,5</i>

Ievaddatu un izvaddatu failu nosaukumi jānorāda **bez** pilnā ceļa (uzskatiet, ka tie atrodas tekošajā katalogā) un tieši tā, kā norādīts uzdevuma formulējumā (**ar mazajiem burtiem**).

Izpildes laika atmiņas ierobežojums: **256MB**.

Maksimāli iespējamais punktu skaits par uzdevumu: **100**.

Lai risinājums tiktu atzīts par derīgu pamattestēšanai, tam jāizdod pareiza atbilde **visiem** uzdevuma formulējumā dotajiem **piemēriem**.

Viens un tas pats tests vai testu grupa var atbilst vairākiem apakšuzdevumiem. Ir garantēts, ka visi noteikta apakšuzdevuma testi atbilst šī apakšuzdevuma aprakstā dotajiem ierobežojumiem, bet ne tas, ka visi dotā uzdevuma testi ar šādiem ierobežojumiem ir iekļauti šajā apakšuzdevumā.

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL:

- FreePascal (versija 2.6.2) ar parametriem
-O2 -XS -Sg -Cs64000000

Valodai C:

- GNU C (versija 4.7.3) ar parametriem
-std=gnu99 -O2 -s -static -lm -xc

Valodai C++:

- GNU C++ (versija 4.7.3) ar parametriem
-O2 -s -static -xc++

Valodai Java:

- Java7 (versija OpenJDK 1.7.0_25 jeb 7u25)



Datums

Dažreiz no astoņiem cipariem iespējams izveidot vienu vai vairākus korektus datumus formātā „dd.mm.gggg”. Dienas numuru apzīmē „dd” un tās pieraksts vienmēr sastāv no diviem cipariem, vienciparu skaitļa gadījumā priekšā pierakstot nebūtisko nulli. Mēneša numuru apzīmē „mm” un tā pieraksts vienmēr sastāv no tieši diviem cipariem, vienciparu skaitļa gadījumā priekšā pierakstot nebūtisko nulli. Gada skaitli apzīmē „gggg” un tā pieraksts vienmēr sastāv no tieši četriem cipariem, nepieciešamības gadījumā skaitļa priekšā pierakstot nebūtiskās nulles.

Pieņemsim, ka laikā no 01.01.0001 līdz 31.12.9999 ir darbojies un darbosies tāds kalendārs, kāds tas ir šobrīd. Tas nozīmē, ka katrs gads ir sadalīts divpadsmit mēnešos, kur pirmajā, trešajā, piektajā, septītajā, astotajā, desmitajā un divpadsmitajā mēnesī ir 31, bet ceturtajā, sestajā, devītajā un vienpadsmitajā – 30 dienas. Gada otrajā mēnesī īsajos gados ir 28, bet garajos – 29 dienas. Garie gadi ir tie, kuros gada skaitlis dalās ar 4, izņemot tos, kas dalās ar 100, bet nedalās ar 400.

Pieļaujami ir tikai datumi, kas atbilst aprakstītajam kalendāram, turklāt gadam jābūt robežās no 1 līdz 9999.

Tā no ciparu komplekta 0,2,2,6,9,9,9,9 var izveidot šādus 27 korektus datumus:

29.09.2699, 29.09.2969, 29.09.2996, 29.06.2999, 26.09.2999, 29.09.6299, 29.09.6929,
 29.09.6992, 22.09.6999, 29.09.9269, 29.09.9296, 29.06.9299, 26.09.9299, 29.09.9629,
 29.09.9692, 22.09.9699, 29.09.9926, 29.06.9929, 26.09.9929, 29.09.9962, 22.09.9969,
 29.06.9992, 26.09.9992, 29.02.9996, 22.09.9996, 22.06.9999, 26.02.9999.

Uzrakstiet programmu, kas nosaka, cik dažādus korektus datumus aprakstītajā formātā ir iespējams izveidot no ievadītiem astoņiem cipariem!

Ievaddati. Teksta faila **datums.dat** pirmajā rindā doti astoņi cipari bez atdalošām tukšumzīmēm.

Izvaddati. Teksta faila **datums.rez** vienīgajā rindā jāizvada vesels skaitlis - dažādo korekto datumu skaits, ko iespējams izveidot no dotajiem astoņiem cipariem.

Piemēri.

Ievaddati datums.dat	Izvaddati datums.rez
02269999	27
Atbilst tekstā dotajam piemēram	

Ievaddati datums.dat	Izvaddati datums.rez
29999992	0

1. apakšuzdevuma testu ievaddati.

Ievaddati datums.dat	Ievaddati datums.dat
01010101	12213443

Apakšuzdevumi un to vērtēšana.

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie divi testi	2
2.	Visi dotie ir nenulles cipari	16
3.	Dotajā ciparu komplektā ir tikai divi dažādi	16
4.	Dotajā ciparu komplektā neviens cipars nepārsniedz 3	16
5.	Bez papildus ierobežojumiem	50
Kopā:		100

Bobslejs

Bobsleja sacensību televīzijas pārraižu laikā pēc katras finišējušās komandas tiek parādīts, kurā vietā tā šobrīd atrodas visu finišējušo komandu konkurencē (skat. att.).

Pieņemsim, ka visas bobsleja komandas ir sanumurētas ar naturāliem skaitļiem no 1 līdz N . Tās trasē ir devušās pēc kārtas numuru pieaugšanas secībā un visas veiksmīgi finišējušās, un ir zināms, kurā vietā katra no komandām atradās tūlīt pēc finišēšanas.

Piemēram, septiņu komandu gadījumā, ja parādītās vietas pēc finiša ir 1,1,2,1,3,1,4, tad tas nozīmē, ka pēc otrā brauciena komandu secība bija 2-1, pēc trešā 2-3-1, pēc ceturtā 4-2-3-1, pēc piektā 4-2-5-3-1, pēc sestā 6-4-2-5-3-1, pēc septītā 6-4-2-7-5-3-1.



Uzrakstiet programmu, kas zināmam komandu skaitam un ieņemtajām vietām tūlīt pēc finišēšanas nosaka galīgo vietu secību sacensību beigās!

Ievaddati. Teksta faila `bobslejs.dat` pirmajā rindā dota naturāla skaitļa N (bobsleja komandu skaits, $N \leq 10^5$) vērtība. Nākamajā faila rindā doti N naturāli skaitļi – komandu ieņemto vietu tūlīt pēc finiša numuri. Katram i ($1 \leq i \leq N$) i -tais skaitlis šajā rindā norāda i -tās komandas ieņemto vietu tūlīt pēc finiša. Starp katriem diviem blakus skaitļiem ievaddatos ir viena tukšumzīme.

Izvaddati. Teksta faila `bobslejs.rez` vienīgajā rindā jāizvada N naturāli skaitļi – komandu numuri tādā secībā, kādā tās ierindojušās sacensību beigās, sākot ar pirmās vietas ieguvēju. Starp katriem diviem blakus skaitļiem izvaddatos jāizvada viena tukšumzīme.

Piemērs (atbilst tekstā dotajam piemēram).

Ievaddati <code>bobslejs.dat</code>	Izvaddati <code>bobslejs.rez</code>
7	6 4 2 7 5 3 1
1 1 2 1 3 1 4	

1. apakšuzdevuma testu ievaddati.

Ievaddati <code>bobslejs.dat</code>
10
1 2 1 2 1 2 1 2 1 2

Ievaddati <code>bobslejs.dat</code>
19
1 2 3 4 3 2 1 2 3 4 3 2 1 2 3 4 3 2 1

Apakšuzdevumi un to vērtēšana.

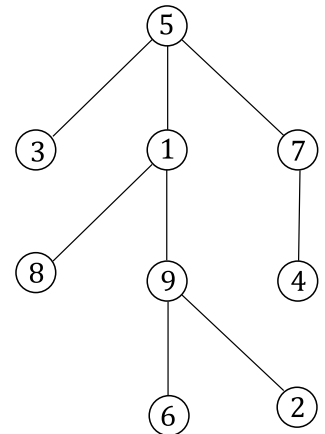
Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie divi testi	2
2.	Nevienas komandas ieņemtā vieta nav lielāka par desmito	12
3.	$N \leq 1000$	16
4.	$N \leq 10000$	30
5.	Bez papildus ierobežojumiem	40
Kopā:		100

Attēli no FIBT vietnes <http://www.fibt.com>



Nederīgie autobusi

Pilsētā ir N autobusu pieturas. Pieturas ir sanumurētas ar naturāliem skaitļiem no 1 līdz N pēc kārtas. Pirmajā pieturā (pieturā ar numuru 1) atrodas pasažieris, kurš jau stundu gaida autobusu, kas dosies uz viņam derīgo pieturu (kas, protams, nav 1.pietura). Autobusu maršruti ir izveidoti tā, ka no 1.pieturas līdz jebkurai citai pieturai ir iespējams nokļūt tikai vienā vienīgā veidā un neviena autobusu maršrutā neviena pietura neatkārtojas. Pasažieris, iekāpjot autobusā 1. pieturā, var aizbraukt līdz atbilstošā maršruta galapunktam vai jebkurai pieturai, kas ir pa ceļam.



Ir zināms, ka neviena autobuss, kas pēdējās stundas laikā aizbraucis no 1.pieturas, pasažierim nederēja. Nepieciešams noteikt, kura (vai kuras) var būt pasažierim derīgās pieturas.

Piemēram, ja deviņas pieturas izvietotas tā, kā redzams zīmējumā, un no 1. pieturas autobusi līdz šim devušies uz 2., 3., 2. un 4. pieturu, tad pasažierim derīgas var būt 6. vai 8. pietura. Uz visām citām pieturām pasažieris būtu varējis aizbraukt ar kādu no jau aizbraukušajiem autobusiem.

Uzrakstiet programmu, kas dotiem jau aizbraukušo autobusu maršrutu galapunktiem nosaka, kuras varētu būt pasažierim derīgās pieturas!

Ievaddati. Teksta faila **autobusi.dat** pirmajā rindā dots pieturu izvietojuma apraksts. Pirmais skaitlis šajā rindā ir naturāls skaitlis $N(2 \leq N \leq 10^5)$, kas apzīmē pieturu skaitu. Nākamie skaitļi norāda pieturu savstarpējo novietojumu. Katram $i(2 \leq i \leq N)$ i -tais skaitlis rindā norāda, kāds ir pēdējās pieturas, kas atrodas pirms pieturas ar numuru i , numurs maršrutā no 1.pieturas līdz pieturai ar numuru i . Faila otrajā rindā dots līdz šim no 1.pieturas aizbraukušo autobusu apraksts. Pirmais skaitlis šajā rindā ir naturāls skaitlis $A(1 \leq A \leq 10^5)$, kas apzīmē aizbraukušo autobusu skaitu. Nākamie A skaitļi norāda šo autobusu maršrutu galapunktu pieturu numurus. Starp katriem diviem blakus skaitļiem ievaddatos ir viena tukšumzīme.

Izvaddati. Teksta faila **autobusi.rez** vienīgajā rindā augošā secībā jāizvada visi iespējamie pasažierim derīgo pieturu numuri. Starp katriem diviem blakus skaitļiem ievaddatos jāatstāj viena tukšumzīme. Zināms ka visiem ievaddatiem, kas tiks izmantoti testēšanai, eksistē vismaz viena derīga pietura.

Piemērs (atbilst tekstā dotajam piemēram).

ievaddati autobusi.dat	Izvaddati autobusi.rez
9 9 5 7 1 9 5 1 1 4 2 3 2 4	6 8

1. apakšuzdevuma testu ievaddati.

ievaddati autobusi.dat
13 11 5 5 2 2 6 11 11 2 1 10 6 7 9 3 13 10 7 8 3

ievaddati autobusi.dat
14 11 12 12 10 1 2 11 12 1 1 1 6 2 6 14 5 4 9 8 7



Apakšuzdevumi un to vērtēšana.

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie divi testi	2
2.	Visas pieturas ir tieši saistītas ar ne vairāk kā divām pieturām	8
3.	Autobusa maršrutā no 1.pieturas līdz jebkurai citai pieturai atrodas ne vairāk kā 20 citas pieturas	10
4.	$A \leq 10$	16
5.	$N \leq 1000$	16
6.	Bez papildus ierobežojumiem	48
	Kopā:	100