

LATVIJAS 21. INFORMĀTIKAS OLIMPIĀDES
IV POSMA UZDEVUMU APSKATS
Pirmā diena (2008. gada 3. aprīlis)



Uzdevuma nosaukums:	NEDILSTOŠAS VIRKNES	MEGAKASTES	MAKSIMĀLĀ XOR SUMMA
Ievaddatu faila nosaukums:	nv.dat	kastes.dat	maxor.dat
Izvaddatu faila nosaukums:	nv.rez	kastes.rez	maxor.rez
Izpildes laika ierobežojums vienam testpiemēram (laiks tiek mērīts uz testēšanas servera):	0,25 sekundes	2 sekundes	0,1 sekunde
Atmiņas ierobežojums:	64MB	64MB	64MB
Maksimāli iespējamais punktu skaits par uzdevumu:	100	100	100
Nosacījums, lai testēšanas serveris atzītu programmu par derīgu testēšanai:	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem.	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem.	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem.

Ievaddatu un izvaddatu failus norādiet **bez** pilnā ceļa (uzskatiet, ka tie atrodas tekošajā katalogā) un tieši tā, kā norādīts uzdevuma formulējumā (**ar mazajiem burtiem**)!

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL:

- FreePascal (versija 2.2.0) ar parametriem `-O2 -XS`

Valodai C:

- GNU C (versija 3.4.2) ar parametriem `-std=c99 -O2 -s -static -lm`
- Microsoft Visual C 2005 ar parametriem `/TC /O2`

Valodai C++:

- GNU C++ (versija 3.4.2) ar parametriem `-O2 -s -static`
- Microsoft Visual C++ 2005 ar parametriem `/TP /O2`

Katra uzdevuma pēdējo akceptēto programmas kodu pēc nosūtīšanas uz serveri saglabāriet arī darba datorā un neizdzēsiet pēc sacensību beigām!

Faila operāciju (datu ielasīšanas un izvadīšanas) ātrdarbības testēšanai uz servera var izmantot šim mērķim speciāli izveidoto uzdevumu “**Ātrdarbības tests**”. Vienīgā piemēra ievaddatu fails `tests.dat` satur 10^6 rindu, i-tajā faila rindā dots skaitlis i ($1 \leq i \leq 10^6$). Atbilde tiek uzskatīta par pareizu, ja iesūtītā programma izveido identisku izvaddatu failu `tests.rez`. Laika limits testa izpildei – 5 sekundes.

1. “NEDILSTOŠAS VIRKNES”

Uz tāfeles uzrakstīta naturālu skaitļu virkne. Vienā gājienā atļauts izvēlēties un nodzēst dažus no uzrakstītajiem skaitļiem, ja katram izvēlētajam skaitlim ir spēkā šāda īpašība: visi izvēlētie skaitļi, kas uzrakstīti pa kreisi no tā, ir mazāki vai vienādi ar to. Katrā gājienā vismaz viens skaitlis ir jānodzēš.

Piemēram, ja sākumā uz tāfeles bija uzrakstīta virkne 3, 1, 1, 5, 9, 5, 7, tad pirmajā gājienā drīkst izvēlēties un nodzēst abus vieniniekus, iegūstot virkni 3, 5, 9, 5, 7. Otrajā gājienā var nodzēst devītnieku, iegūstot virkni 3, 5, 5, 7. Trešajā gājienā var nodzēst visus atlikušos skaitļus. Tomēr šī gājienu virkne nav īsākā iespējamā – dotajam piemēram sākotnējo virkni ir iespējams pilnībā nodzēst, izmantojot tikai divus gājienu.

Uzrakstiet programmu, kas dotai virknei nosaka, pēc kāda mazākā gājienu skaita iespējams nodzēst visu virkni, kā arī atrod atbilstošos gājienu!

Ievaddati

Teksta faila `nv.dat` pirmajā rindā dots naturāls skaitlis N – uz tāfeles uzrakstītās skaitļu virknes garums ($N \leq 200\,000$).

Nākamajās N faila rindās doti virknes locekļi – naturāli skaitļi, kas mazāki par 2^{31} . Katrā faila rindā ir dots viens skaitlis. Virknes i -tais loceklis dots faila $(i+1)$ -ajā rindā ($1 \leq i \leq N$).

Izvaddati

Teksta faila `nv.rez` pirmajā rindā jāizvada naturāls skaitlis – mazākais gājienu skaits, kāds nepieciešams, lai nodzēstu visu doto virkni.

Nākamajās N faila rindās jāizvada informācija par to, kurā gājienā ir jānodzēš katrs no virknes locekļiem. $(i+1)$ -ajā rindā jāizvada naturāls skaitlis – gājiena numurs, kurā jānodzēš virknes i -tais loceklis ($1 \leq i \leq N$).

Ja iespējami vairāki atrisinājumi, jāizvada jebkurš viens no tiem.

Piemēri

Ievaddati (fails <code>nv.dat</code>)	Izvaddati (fails <code>nv.rez</code>)	Piezīmes
4	2	Der arī atrisinājums:
13	1	
11	2	
14	1	
12	2	
		1
7	2	Der arī citi atrisinājumi.
3	1	
1	2	
1	2	
5	1	
9	1	
5	2	
7	2	

Vērtēšana

Daļēju punktu skaitu saņems arī risinājums, kas pareizi atrod tikai mazāko gājienu skaitu (pareiza pirmā izvaddatu faila rinda).

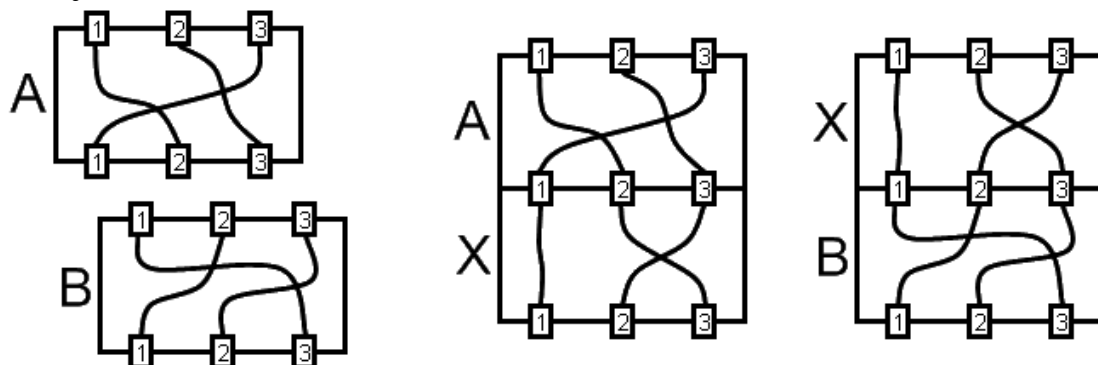
2. “MEGAKASTES”

Lai nopelnītu naudu atpūtas braucienam uz Ēģipti, pensionētā skolotāja Alma ir pievērsusies ienesīgam biznesam – megakastu būvēšanai.

Megakaste tiek būvēta no divām parastām kastēm. Parastajām kastēm ir N ieejas un N izejas, kas sanumurētas ar skaitļiem no 1 līdz N pēc kārtas, bet kastes iekšienē ir N vadi, kas savieno ieejas ar izejām tā, ka katra ieeja ir savienota ar tieši vienu izeju un otrādi. Megakasti būvē, saliekot divas parastas kastes virknē un savienojot pirmās kastes izejas ar otrās kastes ieejām (i -to izeju ar i -to ieeju, $1 \leq i \leq N$). Megakastēm ir spēkā tāda pati īpašība kā parastajām kastēm: katra ieeja ir savienota ar tieši vienu izeju un otrādi.

Almas dzimtās pilsētas iedzīvotāji ir ļoti pesimistiski, tāpēc visas preces tiek tirgotas un pirktas divos eksemplāros.

Bēniņos Almai ir izveidojusies liela parasto kastu kolekcija. Iedzīvotāji par megakastēm ir gatavi maksāt krietni lielāku naudas summu nekā par parastajām kastēm, tāpēc skolotāja ir izprātojusi, ka vislabāk nopelnīt izdosies šādi: ik dienas viņa no bēniņiem atnes kaut kādas divas parastas kastes A un B un tad izspriež, kādas divas vienādas parastas kastes X papildus ir jāiegādājas, lai, saliekot virknē kastes A un X (tieši šādā secībā), kā arī kastes X un B (tieši šādā secībā), iegūtu divas vienādas megakastes (divas kastes ir vienādas, ja to ieejas ar vienādiem numuriem ir savienotas ar izejām ar vienādiem numuriem).



Piemēram, Alma no kastēm A un B (skat. zīmējumu), var uzbūvēt divas vienādas megakastes, iegādājoties divas vienādas kastes X, kurām pirmā ieeja ir savienota ar pirmo izeju, otrā ieeja – ar trešo izeju, bet trešā ieeja – ar otro izeju.

Alma ir nogurusi katru dienu gudrot, kādas kastes X ir jāiegādājas, tāpēc viņa lūdz Jūsu palīdzību.

Uzrakstiet programmu, kas nosaka, kādas kastes X skolotājai ir jāiegādājas!

Ievaddati

Teksta faila `kastes.dat` pirmajā rindā dota naturāla skaitļa N ($N \leq 10^6$) vērtība – vadu skaits katrā no kastēm A un B.

Nākamajā faila rindā dots kastes A apraksts – N naturāli skaitļi a_1, a_2, \dots, a_N . Skaitlis a_i ir izejas numurs, ar kuru ir savienota kastes A i -tā ieeja ($1 \leq i \leq N$). Visi šie skaitļi ir atšķirīgi, un neviens no tiem nepārsniedz N .

Nākamajā faila rindā dots kastes B apraksts šādā pašā formātā.

Katri divi blakusesoši skaitļi failā atdalīti ar tukšumzīmi.

Izvaddati

Teksta faila `kastes.rez` vienīgajā rindā jāizvada kastes X apraksts tādā pašā formātā, kā kastes ir dotas ievaddatu failā. Ja iespējami vairāki atrisinājumi, jāizvada jebkurš viens no tiem. Ja kaste X ar vajadzīgo īpašību neeksistē, faila vienīgajā rindā jāizvada skaitlis 0.

Piemēri

Ievaddati (fails kastes.dat)	Izvaddati (fails kastes.rez)	Piezīmes
3 2 3 1 3 1 2	1 3 2	Der arī citi atrisinājumi, piemēram: 2 1 3
4 3 1 4 2 3 1 4 2	1 2 3 4	Der arī citi atrisinājumi.
3 1 2 3 3 2 1	0	

3. "MAKSIMĀLĀ XOR SUMMA"

Lai aprēķinātu divu vai vairāku bināru skaitļu *XOR summu*, nepieciešams katrai pozīcijai aprēķināt tajā ierakstīto bitu summu. Ja bitu summa ir nepāra skaitlis, tad rezultāta atbilstošā bita vērtība ir 1. Pretējā gadījumā šī bita vērtība ir 0.

Attēlā parādīta skaitļu 10111, 10010 un 00011 XOR summas aprēķināšana:

$$\begin{array}{r} 10111 \\ 10010 \\ 00011 \\ \hline \text{XOR summa } 00110 \end{array}$$

Doti N veseli nenegatīvi K -ciparu bināri skaitļi A_1, A_2, \dots, A_N . No tiem ir jāizvēlas daži skaitļi (vismaz viens) tā, lai izvēlēto skaitļu XOR summa būtu lielākā iespējamā (viena skaitļa XOR summa sakrīt ar šo pašu skaitli). Var izvēlēties arī visus dotos skaitļus.

Uzrakstiet programmu, kas dotiem skaitļiem atrod lielāko iespējamo XOR summu, kādu var iegūt, izvēloties vienu vai vairākus no dotajiem skaitļiem!

Ievaddati

Teksta faila `maxor.dat` pirmajā rindā dotas naturālu skaitļu N un K ($N, K \leq 250$) vērtības.

Katrā no nākamajām N faila rindām dots viens vesels nenegatīvs binārs skaitlis, kas pierakstīts ar tieši K cipariem (skaitļa sākumā var būt nebūtiskas nulles). Katram i ($1 \leq i \leq N$) A_i vērtība dota faila $(i+1)$ -ajā rindā.

Izvaddati

Teksta faila `maxor.rez` vienīgajā rindā jāizvada K -ciparu binārs skaitlis – lielākā iespējamā XOR summa, kādu var iegūt, izvēloties dažus no dotajiem skaitļiem.

Piemēri

Ievaddati (fails maxor.dat)	Izvaddati (fails maxor.rez)	Piezīmes
3 4 0001 0110 1000	1111	Visu trīs skaitļu XOR summa.
3 5 00111 01001 00111	01110	00111 un 01001 XOR summa.
2 2 11 01	11	11 XOR summa.