LATVIJAS 23.INFORMĀTIKAS OLIMPIĀDES III POSMS
VECĀKĀS GRUPAS UZDEVUMU APSKATS
Pirmā diena (2010.gada 10.marts)

Uzdevuma nosaukums:	UNV PROTOKOLS	ŪDENS NĀK!	ŠAHA TORŅI
Ievaddatu faila nosaukums:	`protokols.dat`	`udens.dat`	`torni.dat`
Izvaddatu faila nosaukums:	`protokols.rez`	`udens.rez`	`torni.rez`
Izpildes laika ierobežojums vienam testpiemēram (laiks tiek mērīts uz testēšanas servera):	0,2 sekundes	1 sekunde	1,5 sekundes
Atmiņas ierobežojums:	64MB	64MB	64MB
Steka atmiņas ierobežojums:	16MB	16MB	16MB
Maksimāli iespējamais punktu skaits par uzdevumu:	100	100	100

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 –Sg –Cs16777216`

Valodai C:

- GNU C (versija 3.4.2 un 4.4.1) ar parametriem
  `-std=c99 -O2 -s -static –lm –Wl,--stack,16777216`
- Microsoft Visual C 2008 ar parametriem
  `/TC /O2 /link /STACK:16777216`

Valodai C++:

- GNU C++ (versija 3.4.2 un 4.4.1) ar parametriem
  `-O2 -s –static –Wl,--stack,16777216`
- Microsoft Visual C++ 2008 ar parametriem
  `/TP /O2 /link /STACK:16777216`

UNV PROTOKOLS

Hokeja speciālisti un līdzjutēji jau sen ir ievērojuši, ka ne visi gūtie vārti ir vienlīdz vērtīgi. Viens no terminiem, kas radies šo novērojumu laikā, ir „**uzvaru nesošie vārti**” (jeb, saīsināti, UNV). Uzvaru nesošos vārtus nosaka pēc katra vārtu guvuma. Tie ir vārti, kas, ja spēle būtu beigusies šajā brīdī, būtu noteikuši uzvarētāju. Ja kādā brīdī rezultāts ir neizšķirts, tad tobrīd nevieni vārti nav uzvaru nesoši.

Piemēram, ja savā starpā spēlēja komanda A un B un vārtus guva šādā secībā: AABBBB, tad pēc katra vārtu guvuma nosakot uzvaru nesošos vārtus, mēs iegūtu:

Gūto vārtu kārtas numurs	Vārtu guvēja	Rezultāts		Uzvaru nesošo vārtu numurs	Piezīme
		A	B		
1	A	1	0	1	
2	A	2	0	1	Uzvarai būtu pieticis arī ar 1:0 (pirmajiem gūtajiem vārtiem)
3	B	2	1	2	
4	B	2	2	0	Neizšķirta gadījumā nevieni vārti nav uzvaru nesoši – apzīmēsim to ar 0 kā uzvaru nesošo vārtu numuru.
5	B	2	3	5	
6	B	2	4	5	Uzvarai būtu pieticis arī ar 2:3 (5.vārti)

Šajā gadījumā uzvaru nesošo vārtu numuru virkne jeb **UNV protokols** ir
1 1 2 0 5 5.

Tieši tāds pats UNV protokols būtu arī vārtu gūšanas secībai BBAAAA.

Savukārt, ja spēles gaita būtu bijusi BBBAB vai AAABA, tad UNV protokols būtu
1 1 1 2 2.

Interesants ir arī pretējais uzdevums – zinot UNV protokolu un pieņemot, ka savā starpā spēlēja komandas A un B, noteikt vārtu gūšanas secību.

Diemžēl, UNV protokolā var būt ielaista kāda kļūda, kas neļauj korekti atjaunot spēles gaitu. Šādā gadījumā jācenšas spēles gaitu atjaunot cik tālu iespējams.

Uzrakstiet programmu, kas ievadītam UNV protokolam izvada spēles gaitā gūto vārtu virkni pareizā secībā!

Ievaddati

Teksta faila **protokols.dat** pirmajā rindā dota naturāla skaitļa N(N ≤ 100 000) vērtība – UNV protokola garums. Faila otrajā rindā dots UNV protokols – veselu nenegatīvu skaitļu virkne garumā N. Katri divi blakus skaitļi ievaddatos ir atdalīti ar tukšumzīmi. Katram i(1 ≤ i ≤ N) i-tais skaitlis virknē ir uzvaru nesošo vārtu numurs pēc i-to vārtu guvuma spēlē vai 0, ja tobrīd rezultāts ir neizšķirts.

Izvaddati

Teksta failam **protokols.rez** jāsatur viena rinda, kurā izvadīta simbolu virkne garumā N – vārtu guvumu apraksts hronoloģiskā secībā. Katram i(1 ≤ i ≤ N) i-tais simbols virknē ir tās komandas burts(A vai B), kas guvusi i-tos vārtus vai zvaigznīte(*), ja šajā vietā protokols jau ir atzīts par nekorektu. Tiklīdz izvaddatos parādās zvaigznīte, tas nozīmē, ka visiem pārējiem simboliem virknē līdz pat virknes beigām arī ir jābūt zvaigznītēm (vienā brīdī nekorekts protokols vairs nevar kļūt korekts). Ja dotajam UNV protokolam atbilst vairākas vārtu guvumu virknes, tad jāizvada jebkura no tām. Ja dotajam UNV protokolam neviena korekta vārtu guvumu virkne neatbilst, tad jāizvada tā virkne (vai viena no tādām virknēm), kurā korekti aizpildītā virknes daļa ir pēc iespējas lielāka.

Piemēri

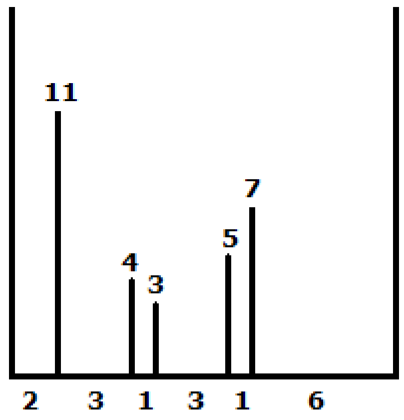
Ievaddati (protokols.dat)	Izvaddati(protokols.rez)	Piezīmes
6 1 1 2 0 5 5	AABBBB	Atbilst formulējumā dotajam piemēram. Pareiza izvaddatu virkne būtu arī BBAAAA, AABBAA vai BBAABB.

Ievaddati (protokols.dat)	Izvaddati(protokols.rez)	Piezīmes
5 1 2 3 3 3	A****	Pēc otro vārtu guvuma otrie vārti nevarēja būt uzvaru nesoši. Ja otros vārtus guva A, tad tādiem bija jābūt pirmajiem vārtiem. Ja otros vārtus būtu guvusi komanda B, tad tobrīd rezultāts būtu neizšķirts un UNV protokolā otrajam skaitlim bija jābūt nullei. Pareiza izvaddatu virkne būtu arī B****.

Ievaddati (protokols.dat)	Izvaddati(protokols.rez)	Piezīmes
3 1 0 3	ABB	Arī ABA, BAA, BAB būtu bijušas derīgas vārtu guvumu virknes.

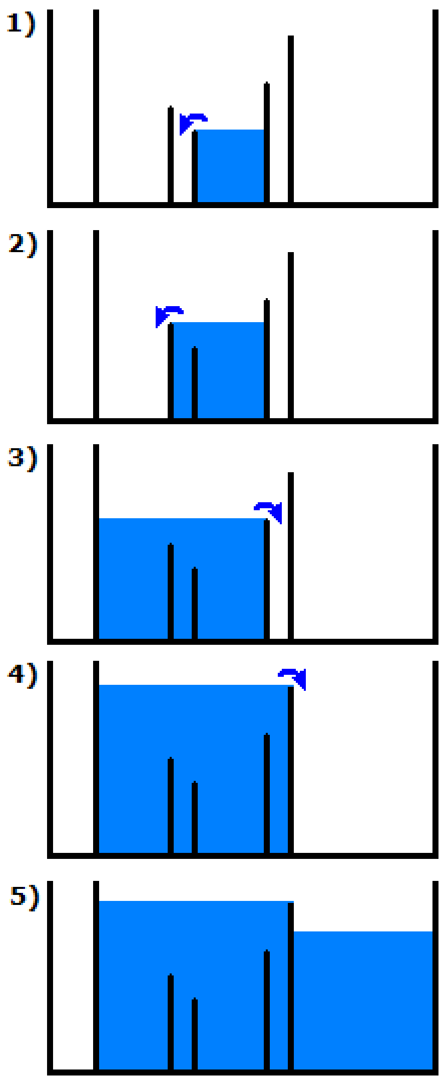
ŪDENS NĀK!

Taisnstūra paralēlskaldņa formas ūdens tvertne šķērsgriezumā izskatās kā taisnstūris bez augšējās malas. Tvertne ar *šķērssienu* palīdzību ir sadalīta *nodalījumos*. Nodalījumi tiek numurēti ar naturāliem skaitļiem no 1 pēc kārtas, sākot no kreisās puses uz labo. Katra nodalījuma platums un katras šķērssienas augstums ir izsakāms veselā skaitā vienību. Tvertnes trešā dimensija ir tāda, ka 1x1 vienībai šķērsgriezumā dabā atbilst vienu litru liels tilpums. Piemēram, šķērssienas tvertnē var būt izvietotas tā, kā redzams zīmējumā.



Skaitlis pie katras šķērssienas norāda tās augstumu, bet skaitlis zem attiecīgā nodalījuma – tā platumu. Zināms, ka nav divu šķērssienu, kuru augstums būtu vienāds. Tvertnes sānu malas ir pietiekami augstas, lai tvertnē varētu iepildīt jebkuru ūdens daudzumu. Sākumā tvertne ir tukša. Ūdens iepildīšana tvertnē notiek no krāna, kas atrodas virs kāda viena nodalījuma, un tvertnē vienmēr tiek iepildīts ūdens daudzums, kas izsakāms veselos litros. Pieņemsim, ka šoreiz krāns atrodas virs ceturtā nodalījuma un tvertnē tiks iepildīti 90 litri ūdens. Aplūkosim šo procesu sīkāk:

1) pēc deviņu litru iepildīšanas ceturtais nodalījums būs piepildīts un ūdens pāri šķērssienai sāks ieplūst trešajā nodalījumā;
2) pēc 16 litru iepildīšanas būs piepildīts gan ceturtais, gan trešais nodalījums, un ūdens sāks ieplūst otrajā nodalījumā;
3) pēc 35 litru iepildīšanas būs piepildīts otrais, trešais un ceturtais nodalījums, un ūdens sāks ieplūst piektajā nodalījumā;
4) pēc 56 litru iepildīšanas otrais līdz piektais nodalījums būs piepildīts un ūdens sāks ieplūst sestajā nodalījumā.
5) pēc 90 litru iepildīšanas sestajā nodalījumā būs 34 litri ūdens.



Tātad iepildīšanas beigās pirmais nodalījums joprojām būs tukšs, otrajā nodalījumā būs 21, trešajā – 7, ceturtajā – 21, piektajā – 7 un sestajā – 34 litri ūdens.

Uzrakstiet programmu, kas dotam tvertnes konfigurācijas un iepildīšanas procesa aprakstam nosaka, kāds ūdens daudzums katrā no nodalījumiem būs iepildīšanas procesa beigās!

Ievaddati

Teksta faila **udens.dat** pirmajā rindā dotas trīs naturālu skaitļu N (nodalījumu skaits, $1 \le N \le 100\,000$), P (nodalījuma, virs kura atrodas iepildes krāns, kārtas numurs; $1 \le P \le N$) un T (iepildāmā ūdens daudzums litros, $1 \le T \le 10^9$) vērtības. Katri divi blakusesoši skaitļi atdalīti ar tukšumzīmi. Katrā no nākamajām N faila rindām dots viena nodalījuma platums – naturāls skaitlis, kura vērtība nepārsniedz 10^9. Katram i ($1 \le i \le N$) i-tā nodalījuma platums dots faila (i+1)-ajā rindā. Katrā no nākamajām N-1 faila rindām dots vienas šķērssienas augstums – naturāls skaitlis, kura vērtība nepārsniedz 10^9. Katram i ($1 \le i < N$) šķērssienas starp i-to un (i+1)-o nodalījumu augstums ir dots faila (N+1+i)-tajā rindā. Zināms, ka visu šķērssienu augstumi ir savā starpā atšķirīgi.

Izvaddati

Teksta failam **udens.rez** jāsatur tieši N rindas. Katram i ($1 \le i \le N$) faila i-tajā rindā jāizvada vesels nenegatīvs skaitlis p_i un naturāls skaitlis q_i, kas atdalīti ar tukšumzīmi. p_i/q_i – ūdens daudzums litros i-tajā nodalījumā. p_i un q_i jābūt savstarpējiem pirmskaitļiem (to lielākajam kopīgajam dalītājam ir jābūt 1).

Ja i-tais nodalījums ir tukšs, tad p_i=0 un q_i=1.

Piemēri

Ievaddati (udens.dat)	Izvaddati (udens.rez)	Piezīme
6 4 90 2 3 1 3 1 6 11 4 3 5 7	**0 1 21 1 7 1 21 1 7 1 34 1**	Atbilst uzdevuma tekstā dotajam piemēram.

Ievaddati (udens.dat)	Izvaddati (udens.rez)
2 1 13 1 7 1	**13 8 91 8**

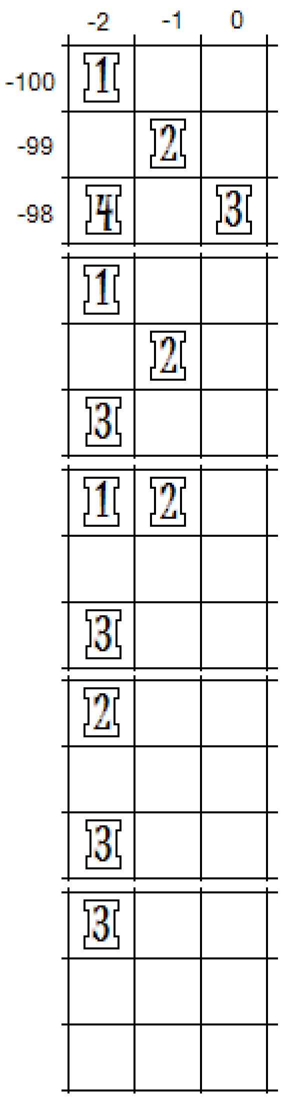
ŠAHA TORŅI

Uz bezgalīgi liela rūtiņu laukuma novietoti K šaha torņi. Gan rindas, gan kolonnas ir numurētas ar veseliem skaitļiem pēc kārtas. Katrs tornis atrodas savā rūtiņā. Divi torņi apdraud (var nosist) viens otru, ja tie atrodas uz vienas horizontāles vai vertikāles un ja starp šiem torņiem nav citu torņu.

Katrā *gājienā* kāds tornis pārvietojas no savas rūtiņas uz kādu citu tajā pašā rindā vai kolonnā. Ja šajā beigu rūtiņā atrodas cits tornis, tad tas tiek *nosists* - t.i., ja tornis A pārvietojas uz lauciņu, ko aizņēma tornis B, tad tornis B tiek noņemts no laukuma un tā vietā tiek novietots tornis A. Uzdevums ir ar pēc iespējas mazāku gājienu skaitu panākt situāciju, ka laukumā paliek viens tornis.

Zīmējumā parādīts piemērs, kur pēc četru gājienu virknes no sākumā esošajiem četriem torņiem laukumā paliek viens.

Uzrakstiet programmu, kas atrod un izvada tādu gājienu secību, ka beigās uz laukuma paliek viens tornis un kopējais gājienu skaits ir mazākais iespējamais!



Ievaddati

Teksta faila **torni.dat** pirmajā rindā dota naturāla skaitļa $K (K \leq 100\,000)$ vērtība. Katrā no nākošajām K faila rindām doti divi veseli skaitļi, kas atdalīti ar tukšumzīmi. Katram $i (1 \leq i \leq K)$ faila i+1-ajā rindā dotas i-tā torņa koordinātas – rindas $r_i(-10^9 \leq r_i \leq 10^9)$ un kolonnas $k_i(-10^9 \leq k_i \leq 10^9)$ numurs.

Izvaddati

Teksta faila **torni.rez** pirmajā rindā jāizvada vesels nenegatīvs skaitlis - mazākais gājienu skaits M, kāds jāizdara, lai uz laukuma paliktu viens tornis. Katrai no nākošajām M faila rindām jāapraksta viens torņa gājiens un jāsatur četri veseli skaitļi, kur katri divi blakus skaitļi atdalīti ar tukšumzīmi. Pirmajiem diviem skaitļiem jābūt torņa sākuma rūtiņas koordinātām (rindas un kolonnas numuram), bet otriem diviem – beigu rūtiņas koordinātām. Katram $i (1 \leq i \leq M)$ faila i+1-ajā rindā jāizvada informācija par i-to gājienu pēc kārtas. Ja iespējamas vairākas derīgas gājienu virknes, izvadiet jebkuru no tām.

Piemērs *(atbilst dotajam zīmējumam)*

Ievaddati (torni.dat)	Izvaddati(torni.rez)

`4`
`-100 -2`
`-99 -1`
`-98 0`
`-98 -2`
	`4`
	`-98 0 -98 -2`
	`-99 -1 -100 -1`
	`-100 -1 -100 -2`
	`-98 -2 -100 -2`