LATVIJAS 22. INFORMĀTIKAS OLIMPIĀDES
ATLASES KĀRTAS UZDEVUMU APSKATS
Pirmā diena (2009. gada 26. marts)

Uzdevuma nosaukums:	**IEKAVAS**	**KVADRĀTI**	**KOKS**
Ievaddatu datnes nosaukums:	`iekavas.dat`	`kvadrati.dat`	`koks.dat`
Izvaddatu datnes nosaukums:	`iekavas.rez`	`kvadrati.rez`	`koks.rez`
Izpildes laika ierobežojums vienam testpiemēram (laiks tiek mērīts uz testēšanas servera):	0,2 sekundes	2 sekundes	0,75 sekundes
Atmiņas ierobežojums:	8MB	32MB	32MB
Maksimāli iespējamais punktu skaits par uzdevumu:	100	100	100

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

Lai iesūtītais risinājums tiktu pieņemts tālākai testēšanai, tam pareizi jāstrādā uz **visiem** uzdevuma formulējumā dotajiem testpiemēriem. Testēšanas serverī noklikšķinot uz iesūtījuma, parādās rezultāts katram testpiemēram tādā pašā secībā, kā tie doti uzdevuma formulējumā.

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL:
- FreePascal (versija 2.2.0) ar parametriem `-O2 -Sg`

Valodai C:
- GNU C (versija 3.4.2) ar parametriem
  `-std=c99 -O2 -s -static -lm`
- Microsoft Visual C 2008 ar parametriem `/TC /O2`

Valodai C++:
- GNU C++ (versija 3.4.2) ar parametriem
  `-O2 -s -static`
- Microsoft Visual C++ 2008 ar parametriem `/TP /O2`

Programmas tiks testētas uz datora ar *Intel® Pentium® 4* 2GHz procesoru.

LATVIJAS 22. INFORMĀTIKAS OLIMPIĀDES ATLASES KĀRTAS UZDEVUMI
Pirmā diena (2009. gada 26. marts)

1. "IEKAVAS"

Par *korektu iekavu virkni* sauc:

1) virkni, kas sastāv no atverošās un aizverošās iekavas: ( )
2) virkni, ko veido divas korektas iekavu virknes, kas uzrakstītas tieši viena aiz otras
3) virkni, ko veido atverošā iekava, korekta iekavu virkne un aizverošā iekava, kas uzrakstītas viena aiz otras.

Piemēram, (( ) ( ) (( ))) ( ) ir korekta iekavu virkne, bet) ( ) (– nav.

Katru korektu iekavu virkni var izveidot no virknes ( ), ja tajā atļauts pēc kārtas noteiktā pozīcijā iespraust atverošās un aizverošās iekavas pāri: ( ). Uzskatīsim, ka ir atļauta tikai viena veida darbība: "iespraust ( ) līdz šim izveidotajā virknē **pirms** i-tā simbola", kur i ir naturāls skaitlis robežās no 1 līdz G+1, kur ar G apzīmēts virknes garums pirms iespraušanas. Ja i = G+1, tad tas nozīmē, ka iekavu pāris tiek pierakstīts līdz šim izveidotās virknes beigās.

Izpildīto darbību virkni var pierakstīt, katrā gājienā pierakstot tikai attiecīgo i vērtību.

Piemēram, darbību virkne "2, 3, 1, 5" apraksta virknes ( ) ((( ) ( ))) veidošanas procesu (kārtējā gājienā iespraustais iekavu pāris dots treknrakstā): sākumā ir dots iekavu pāris ( ), pirmajā gājienā tiek izveidota iekavu virkne (**( )**), otrajā – ((**( )**)), trešajā – **( )** ((( ))), ceturtajā – ( ) ((**( )** ( ))).

Iespējams, ka vienu un to pašu virkni apraksta dažādas darbību virknes. Piemēram, aplūkoto virkni ir iespējams iegūt, izpildot darbību virkni "3, 4, 5, 7". Darbību virknes, kas apraksta vienas un tās pašas iekavu virknes veidošanas procesu, iespējams sakārtot leksikogrāfiskā secībā.

Teiksim, ka darbību virkne $a_1, a_2, a_3, \ldots, a_k$ ir *leksikogrāfiski mazāka* par darbību virkni $b_1, b_2, b_3, \ldots, b_k$, ja ir iespējams atrast tādu j ($1 \le j \le k$), ka $a_j < b_j$, bet visiem i ($1 \le i < j$) izpildās $a_i = b_i$.

Uzrakstiet programmu, kas dotai korektai iekavu virknei atrod darbību virkni, kas apraksta dotās iekavu virknes veidošanas gaitu un ir leksikogrāfiski vismazākā!

Ievaddati

Teksta datnes `iekavas.dat` pirmajā rindā dots naturāls pāra skaitlis N (iekavu virknes garums, $2 < N \le 100\,000$).

Otrajā datnes rindā dota korekta iekavu virkne, kas satur tieši N apaļās iekavas.

Izvaddati

Teksta datnei `iekavas.rez` jāsatur (N/2)−1 rinda. Katram i ($1 \le i < N/2$) i-tajā rindā jāizvada viens naturāls skaitlis – tās pozīcijas, pirms kuras i-tajā gājienā jāiesprauž iekavu pāris, numurs.

Piemērs

Ievaddati	**Izvaddati**	**Piezīmes**

`10`
`()((()()))`
	`1`
	`4`
	`5`
	`5`
		Dotā iekavu virkne tiek iegūta šādi:
		`() -> ()() -> ()(()) -> ()((())) -> ()((()()))`

2. “KVADRĀTI”

Plaknē doti N punkti. Nepieciešams novietot N vienādus kvadrātus ar centriem dotajos punktos tā, ka kvadrātu malas ir paralēlas koordinātu asīm un nekādi divi kvadrāti nepārklājas (taču tie drīkst saskartiem ar malām vai virsotnēm).

Uzrakstiet programmu, kas dotajiem punktiem aprēķina, kāds ir novietojamo kvadrātu lielākais iespējamais malas garums!

Ievaddati

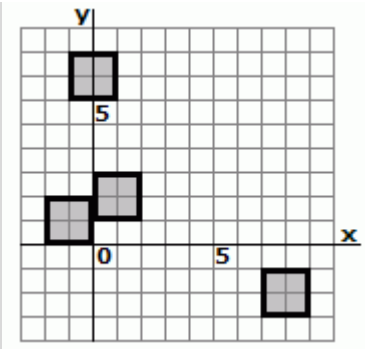
Teksta datnes `kvadrati.dat` pirmajā rindā dots naturāls skaitlis N (punktu skaits, $1 < N \leq 100\,000$).

Katrā no nākamajām N datnes rindām doti divi veseli skaitļi, kas atdalīti ar tukšumzīmi, – viena dotā punkta koordinātas. Punktu koordinātas ir robežās no $-5 \cdot 10^{17}$ līdz $5 \cdot 10^{17}$ ieskaitot. Visi dotie punkti ir atšķirīgi.

Izvaddati

Teksta datnes `kvadrati.rez` vienīgajā rindā jāizvada viens naturāls skaitlis – lielākais iespējamais kvadrātu malas garums.

Piemēri

Ievaddati	**Izvaddati**	
4 -1 1 0 7 1 2 8 -2	2	
2 1 2 4 1	3	

3. “KOKS”

Dots koks, kas satur N virsotnes. Koka virsotnes ir sanumurētas ar naturāliem skaitļiem no 1 līdz N pēc kārtas. Ir atļauts izmest vienu vai vairākas koka šķautnes, neizmetot virsotnes. Rezultātā dotais koks sadalīsies vairākos kokos. Mērķis ir iegūt vismaz vienu koku ar tieši K virsotnēm.

Uzrakstiet programmu, kas dotam kokam un K vērtībai aprēķina, kāds mazākais šķautņu skaits jāizmet, lai vismaz viens no iegūtajiem kokiem saturētu tieši K virsotnes!

Ievaddati

Teksta datnes `koks.dat` pirmajā rindā doti divi naturāli skaitļi N ($N \leq 1\,000$) un K ($K \leq 300$, $K < N$), kas atdalīti ar tukšumzīmi.

Katrā no nākamajām N−1 datnes rindām doti divi atšķirīgi naturāli skaitļi, kas nepārsniedz N, – koka virsotņu, kas savienotas ar šķautni, numuri. Katra šķautne ievaddatos ir dota vienreiz.

Izvaddati

Teksta datnes `koks.rez` vienīgajā rindā jāizvada viens naturāls skaitlis – mazākais šķautņu skaits, kas jāizmet.

Piemēri

Ievaddati	Izvaddati	Piezīmes
5 2 2 1 2 5 3 4 3 2	1	Jāizmet šķautne, kas savieno trešo un otro virsotni.
5 3 3 1 3 2 3 4 3 5	2	Jāizmet jebkuras divas šķautnes.