

**LATVIJAS 20. INFORMĀTIKAS OLIMPIĀDES III POSMA
UZDEVUMU APSKATS
Pirmā diena (2007.gada 19.marts)**



Uzdevuma nosaukums:	IEKAVAS	KOSMOSA PĒTNIEKS	LAMPAS
Ievaddatu faila nosaukums:	iekavas.dat	kosmoss.dat	lampas.dat
Izvaddatu faila nosaukums:	iekavas.rez	kosmoss.rez	lampas.rez
Izpildes laika ierobežojums vienam testam (laiks tiek mērīts uz testēšanas servera):	0,5 sekundes	0,2 sekundes	0,7 sekundes
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 jebkurai četru iekavu virknei	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajam piemēram

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili 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 2.0.4) ar parametriem `-So -Sg -O2 -XS`

Valodai C:

- GNU C (versija 3.4.2) ar parametriem `-x c -static -O2 -lm`
- Microsoft Visual C 2005 ar parametru `/TC /O2`

Valodai C++:

- GNU C++ (versija 3.4.2) ar parametriem `-x c++ -static -O2 -lm`
- Microsoft Visual C++ 2005 ar parametru `/TP /O2`

Jautājumus par uzdevumu formulējumiem uzdodiet sacensību pirmās stundas laikā!

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

LATVIJAS 20. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Pirmā diena (2007.gada 19.marts)



1. "IEKAVAS"

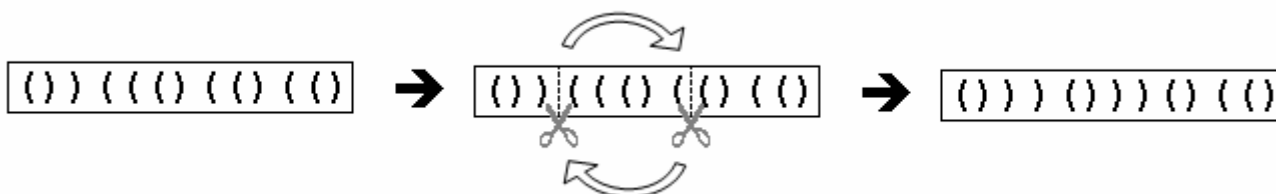
Korektu iekavu izteiksmi definē šādi:

- () ir korekta iekavu izteiksme;
- ja A ir korekta iekavu izteiksme, tad (A) arī ir korekta iekavu izteiksme;
- ja A un B ir korektas iekavu izteiksmes, tad AB arī ir korekta iekavu izteiksme.

Tā, piemēram, ((()()), ((())) un ()()() ir korektas iekavu izteiksmes, bet)(, (), (())(– nav.

Ir dota papīra strēmele, uz kuras uzrakstīta simbolu virkne, kas sastāv tikai no apaļajām iekavām ‘(’ un ‘)’. Vienā gājienā drīkst izvēlēties patvaļīgu strēmeles fragmentu (vienu vai vairākus pēc kārtas sekojošus virknes simbolus), izgriezt to, apgriezt to otrādi un ielīmēt šajā pat vietā. Šī gājiena laikā visas apgriežamā fragmenta iekavas “mainās uz pretējo” – atverošās iekavas kļūst par aizverošajām un aizverošās – par atverošajām. Protams, fragments var atrasties arī strēmeles sākumā vai beigās un tad gājiena izdarīšanai pietiks strēmeli pārgriezt un salīmēt tikai vienā vietā. Ja nepieciešams, arī visu strēmeli var uzskatīt par vienu lielu fragmentu, un apgriezt to otrādi, neizdarot nekādus griezumus un salīmēšanas.

Piemēram, ja sākumā bija uzrakstīta virkne ‘()((()()()’, pēc viena gājiena iespējams iegūt virkni ‘()())()()’, apgriežot fragmentu, kas sākas ar ceturto un beidzas ar astoto virknes simbolu:



Uzrakstiet programmu, kas aprēķina, ar kādu mazāko gājienu skaitu no dotās simbolu virknes iespējams iegūt korektu iekavu izteiksmi, un atrod šos gājienu!

Ievaddati

Teksta faila *iekavas.dat* pirmajā rindā dots naturāls skaitlis N – uz strēmeles uzrakstītās simbolu virknes garums ($N \leq 100\,000$, N – pāra skaitlis). Nākamajā rindā dota simbolu virkne garumā N, kas sastāv tikai no simboliem ‘(’ un ‘)’.

Izvaddati

Teksta faila *iekavas.rez* pirmajā rindā jāizvada vesels nenegatīvs skaitlis M – mazākais gājienu skaits, ar kādu no dotās virknes iespējams iegūt korektu iekavu izteiksmi. Tālākajās M rindās jāizvada šie gājieni, pa vienam katrā rindā. Faila (i+1)-ajā rindā jāizvada i-tā izdarāmā gājiena apraksts – divi veseli pozitīvi skaitļi, kas atdalīti ar tukšumzīmi. Pirmais skaitlis norāda apgriežamā virknes fragmenta pirmā simbola indeksu, bet otrais skaitlis – pēdējā simbola indeksu (indekss ir attiecīgā simbola kārtas numurs gājiena izdarīšanas brīdī, numerāciju sākot no 1). Ja dotajai virknei eksistē vairāki pareizi atrisinājumi, izvadiet jebkuru vienu no tiem!

Piemēri

Ievaddati (<i>iekavas.dat</i>)	Izvaddati (<i>iekavas.rez</i>)	Piezīmes
6 ()((()	0	
6 ()())	1 1 3	Kā faila otrā rinda der arī kāda no šīm rindām: 1 5 2 2 2 4 3 3 3 5

2. "KOSMOSA PĒTNIEKS"

Kosmosa pētnieks Almands jau sen pieraksta no kosmosa nākošos signālus. Katra signāla pierakstīšanai Almands lieto vienu simbolu. Tā kā signāli mēdz atkārtoties, tad vienādo signālu pierakstīšanai vienmēr tiek lietots viens un tas pats simbols, bet dažādu signālu pierakstīšanai tiek lietoti atšķirīgi simboli.

Lai kļūtu slavens kosmosa pētnieku aprindās, Almands cer sagaidīt noteiktu signālu virkni garumā N ($N \geq 1$), ko viņš ir nosaucis par *paroli*. Diemžēl, pēc vairāku gadu desmitu darba (un ļoti garas uztverto signālu virknes pierakstīšanas) Almands ir uzzinājis, ka dažādu signālu izplatīšanās ātrums Visumā var būt dažāds, kas nozīmē to, ka sākumā izsūtīto signālu secība var mainīties. Vēl trakāk – daži signāli to sūtīšanas laikā varēja “sabajāties”, un viena signāla vietā varētu būt uztverts pavisam cits.

Lai tomēr varētu izmantot pierakstīto signālu virkni, Almands ir nolēmis pētīt šīs virknes fragmentus – vienu vai vairākus pēc kārtas sekojošus signālus pierakstītājā signālu virknē.

Almands ir izveidojis divas definīcijas:

- Virknes fragmentu garumā N sauc par *izcilu*, ja, tas vai nu sakrīt ar paroli, vai arī pārkārtojot tajā signālu kārtību, var iegūt paroli.
- Virknes fragmentu garumā N sauc par *labu*, ja, pārkārtojot tajā signālus, var iegūt virkni, kurā ar paroli sakrīt vismaz L ($1 \leq L < N$) signāli.

Skaidrs, ka katrs izcils fragments vienlaicīgi ir arī labs fragments.

Uzrakstiet programmu, kas dotai parolei, L vērtībai un pierakstīto signālu virknei nosaka, cik dažādās vietās pierakstīto signālu virknē atrodami izcilu, un cik vietās - labi fragmenti!

Ievaddati

Teksta faila `kosmoss.dat` pirmajā rindā doti trīs naturāli skaitļi N (paroles garums, $N \leq 10\,000$), L (mazākais simbolu skaits, kādam jāsakrīt, lai fragments tiktu uzskatīts par labu, $1 \leq L < N$) un M (pierakstītās signālu virknes garums, $M \leq 1\,000\,000$). Katri divi blakus skaitļi ir atdalīti ar tukšumzīmi.

Faila otrajā rindā dots paroles pieraksts – simbolu virkne garumā N .

Faila trešajā rindā dota pierakstīto signālu virkne – simbolu virkne garumā M .

Parole un simbolu virkne var saturēt tikai 80 dažādus simbolus, kas doti tabulā:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	P	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	7
8	9	()	{	}	<	>	?	[]	!	@	#	\$	%	&	+	*	=

Izvaddati

Teksta faila `kosmoss.rez` vienīgajā rindā jāizvada divi veseli nenegatīvi skaitļi, kas atdalīti ar tukšumzīmi – dažādo vietu skaits pierakstīto signālu virknē, kur iespējams atrast izcilu fragmentu un dažādo vietu skaits, kur iespējams atrast labu fragmentu.

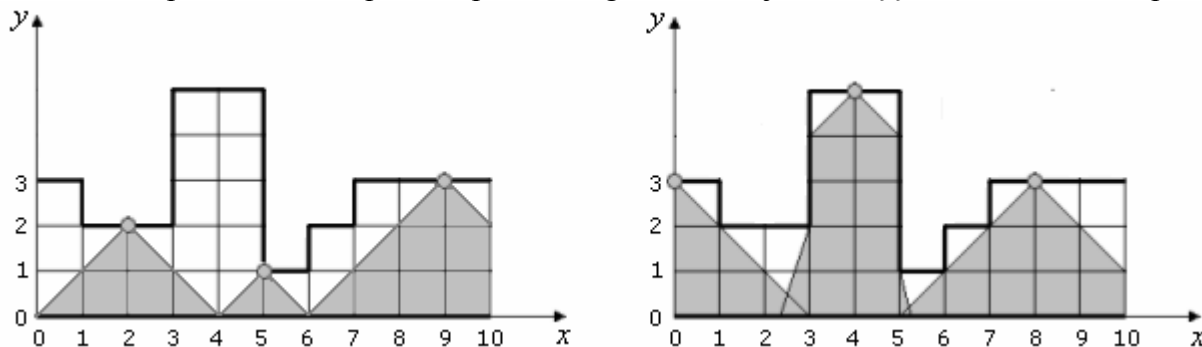
Piemēri

Ievaddati (fails <code>kosmoss.dat</code>)	Izvaddati (fails <code>kosmoss.rez</code>)
5 3 11 abaks abrakadabra	0 7
5 4 11 abaks abrakadabra	0 3
5 4 11 abaks nabagsabaks	2 6

3. "LAMPAS"

Taisnleņķa koordinātu sistēmā ir uzziņmēts telpas sānskats. Grīda ir attēlota ar nogriezni, kas sākas punktā (0,0) un beidzas punktā (N,0). Griesti ir attēloti kā lauza līnija, kuras sākumpunkta x-koordināta ir 0, bet beigu punkta x-koordināta ir N. Tā sastāv no horizontāliem un vertikāliem posmiem, kuru garumi ir veseli skaitļi (pēc horizontāla posma vienmēr seko vertikāls posms un otrādi). Zināms, ka katras lauztās līnijas virsotnes x-koordināta ir lielāka vai vienāda ar iepriekšējās virsotnes x-koordinātu. Visu virsotņu y-koordinātas ir naturāli skaitļi. Pirmais un pēdējais lauztās līnijas posms ir horizontāls.

Jebkura griestu horizontālā posma punktus ar veselām koordinātām (ieskaitot tā galapunktus) iespējams novietot punktveida lampu. Lampa izstaro gaismu uz leju 45° leņķī uz kreiso un labo pusi.



Zīmējumos parādīti divi atšķirīgi visas grīdas apgaismošanas varianti izmantojot trīs lampas (grīda un griesti iezīmēti ar biezāku līniju). Zīmējumā pa kreisi lampas novietotas punktos ar koordinātām (2,2), (5,1) un (9,3) un apgaismotie grīdas segmenti ir attiecīgi [0;4], [4;6] un [6;10] (uzskatām, ka kādas lampas apgaismotais grīdas segments vienmēr ietver abus galapunktus). Zīmējumā pa labi lampas novietotas punktos ar koordinātām (0,3), (4,5) un (8,3). Apgaismotie grīdas segmenti ir attiecīgi [0; 3], [2½; 5¼], [5; 10]. Redzam, ka abos variantos ar šādi novietotām trim lampām visa grīda (nogrieznis [0; 10]) tiek apgaismota.

Uzrakstiet programmu, kas dotai griestu konfigurācijai atrod mazāko iespējamo lampu skaitu, kāds nepieciešams, lai varētu apgaismot visu grīdu!

Ievaddati

Teksta faila `lampas.dat` pirmajā rindā dots vesels pozitīvs skaitlis N ($N \leq 100\,000$) – grīdas garums. Nākamajā rindā doti N naturāli skaitļi. i -tais skaitlis apzīmē griestu augstumu segmentā $[i-1; i]$ katram i ($1 \leq i \leq N$). Neviena skaitļa vērtība nepārsniedz 20000. Katri divi blakusskaitļi atdalīti ar tukšumzīmi.

Izvaddati

Teksta faila `lampas rez` vienīgajā rindā jāizvada vesels skaitlis – mazākais iespējamais lampu skaits, kāds nepieciešams, lai varētu apgaismot visu grīdu.

Piemēri

Ievaddati (fails <code>lampas.dat</code>)	Izvaddati (fails <code>lampas rez</code>)
10 3 2 2 5 5 1 2 3 3 3	3
2 3 1	1
4 1 1 1 1	2

**LATVIJAS 20. INFORMĀTIKAS OLIMPIĀDES III POSMA
UZDEVUMU APSKATS
Otrā diena (2007.gada 20.marts)**



Uzdevuma nosaukums:	BEZMIEGS	PĀRBAUDES PLĀKSNĪTES	KRĀSAINO SKAITĻU STARPĪBA
Ievaddatu faila nosaukums:	bezmiogs.dat	pp.dat	starpiba.dat
Izvaddatu faila nosaukums:	bezmiogs.rez	pp.rez	starpiba.rez
Izpildes laika ierobežojums vienam testam (laiks tiek mērīts uz testēšanas servera):	0,1 sekunde	1,5 sekundes	0,5 sekundes
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 minētajam trīs plāksnīšu, kā arī jebkuram divu plāksnīšu piemēram	Programmai jākompilējas bez kļūdām un jāizdod pareizs rezultāts uzdevuma formulējumā minētajiem piemēriem

Datu un rezultātu failus norādiet **bez** pilnā ceļa (uzskatiet, ka datu un rezultātu faili 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 2.0.4) ar parametriem `-So -Sg -O2 -XS`

Valodai C:

- GNU C (versija 3.4.2) ar parametriem `-x c -static -O2 -lm`
- Microsoft Visual C 2005 ar parametru `/TC /O2`

Valodai C++:

- GNU C++ (versija 3.4.2) ar parametriem `-x c++ -static -O2 -lm`
- Microsoft Visual C++ 2005 ar parametru `/TP /O2`

Jautājumus par uzdevumu formulējumiem uzdodiet sacensību pirmās stundas laikā!

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

LATVIJAS 20. INFORMĀTIKAS OLIMPIĀDES
III POSMA UZDEVUMI
Otrā diena (2007.gada 20.marts)



1. "BEZMIEGS"

Pensionēto skolotāju Almu beidzamajā laikā moka bezmiegs. Nelīdz ne adīšana līdz rīta gaismai, ne klasiskā aitu skaitīšana.

Vienīgais efektīvais līdzeklis ir šāds: Alma uz papīra lapas uzraksta naturālu skaitli N ar vismaz diviem cipariem, aprēķina $N+1$ un uzraksta šī skaitļa ciparus pretējā secībā, nerakstot nebūtiskās nulles (ja tādas ir). Pēc tam iepriekšminētās darbības atkārto ar jauniegūto skaitli, pēc tam ar nākošo, utt. Tā Alma turpina rēķināt un rakstīt līdz iemieg vai arī iegūst viencipara skaitli.

Piemēram, ja sākumā ir uzrakstīts skaitlis 837, tad tas tiek pārveidots šādi: $837 \rightarrow 838 \rightarrow 938 \rightarrow 939 \rightarrow 49 \rightarrow 5$.

Uzrakstiet programmu, kas dotam N nosaka, kādu viencipara skaitli iegūs Alma (protams, ja neiemigs agrāk) un kurš no uzrakstītajiem skaitļiem pēc kārtas tas būs!

Ievaddati

Teksta faila `bezmiegs.dat` pirmajā rindā dots skaitļa N ciparu skaits ($1 < N_ciparu_skaits \leq 1000000$). Faila otrajā rindā dots skaitlis N . Skaitļa N pieraksts nesākas ar 0.

Izvaddati

Teksta faila `bezmiegs.rez` vienīgajā rindā jāizvada vienciparu skaitlis, ko Alma, ja neiemigs, iegūs N pārveidošanas gaitā un kurš no uzrakstītajiem skaitļiem pēc kārtas tas būs (pieņemot, ka skaitļi tiek numurēti pēc kārtas to uzrakstīšanas secībā un pirmajam uzrakstītajam skaitlim ir kārtas numurs 1).

Piemēri

Ievaddati (fails <code>bezmiegs.dat</code>)	Izvaddati (fails <code>bezmiegs.rez</code>)
3 837	5 6
15 8000000000000000	9 128

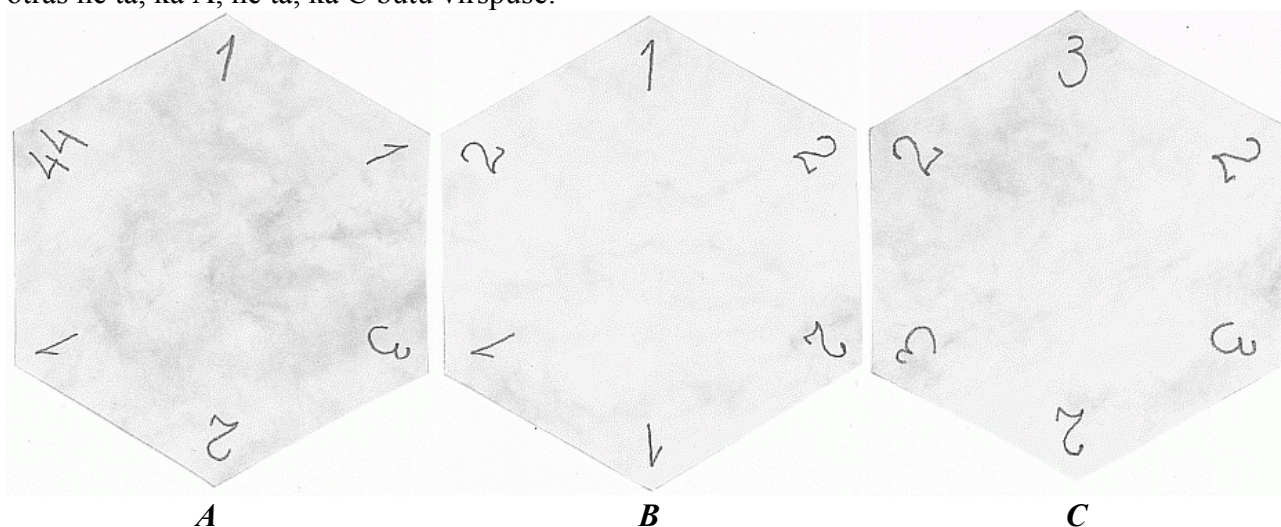
2. "PĀRBAUDES PLĀKSNĪTES"

Arheologi ir atraduši veselu kaudzi ar interesantām vienāda izmēra marmora plāksnītēm. Katra plāksnīte ir regulārs sešstūris, kurai pie katras virsotnes abās plāksnītes pusēs ir pierakstīts viens un tas pats naturāls skaitlis.

Arheologi uzskata, ka šīs plāksnītes ir daļa no slavenajām Karaļa Artūra Augstmaņu Pārbaudes Plāksnītēm.

Šīs pārbaudes būtība bija šāda: augstmanim, kas pretendēja uz kādu amatu, saullēktā tika iedots plāksnīšu komplekts un līdz saulrietam augstmanim bija jāizveido pēc iespējas augstāks plāksnīšu tornis. Plāksnītes drīkstēja likt vienu uz otras tikai tā, ka plāksnīšu virsotnes atrodas viena virs otras un katrai augšējās plāksnītes virsotnei pierakstītais skaitlis nav mazāks par atbilstošajai apakšējās plāksnītes virsotnei pierakstīto skaitli.

Piemēram, ja ir zīmējumā redzamās trīs plāksnītes A, B un C, tad var uzlikt plāksnīti A uz plāksnītes B (iepriekš viena plāksnīte jāapgriež otrādi), C uz B, bet A un C nevar uzlikt vienu uz otras ne tā, ka A, ne tā, ka C būtu virspusē.



Līdz ar to, šādu trīs plāksnīšu gadījumā maksimālais torņa augstums (torni veidojošo plāksnīšu skaits) ir divi.

Uzrakstiet programmu, kas dotam plāksnīšu aprakstam nosaka, kāds ir maksimālā torņa augstums, kādu iespējams izveidot no šīm plāksnītēm!

Ievaddati

Teksta faila `pp.dat` pirmajā rindā dots naturāls skaitlis N (plāksnīšu skaits, $1 < N \leq 3000$).

Katrā no nākošajām N rindām dots vienas plāksnītes apraksts – seši naturāli skaitļi, kādi uzrakstīti pie plāksnītes virsotnēm. Skaitļi doti, apstaigājot virsotnes pulksteņrādītāja virzienā. Starp katriem diviem blakusskaitļiem ir viena tukšumzīme. Neviena pie virsotnes pierakstītā skaitļa vērtība nepārsniedz 1 000 000.

Izvaddati

Teksta faila `pp.rez` vienīgajā rindā jāizvada naturāls skaitlis – lielākais iespējamais plāksnīšu torņa augstums.

Piemērs

Ievaddati (fails <code>pp.dat</code>)	Izvaddati (fails <code>pp.rez</code>)
3	2
1 3 2 1 44 1	
1 2 1 2 2 1	
3 2 3 2 3 2	

3. "KRĀSAINO SKAITĻU STARPĪBA"

Dots vesels pozitīvs skaitlis, kas sastāv no $N(N - \text{pāra skaitlis})$ nenulles cipariem. Katru tā ciparu nokrāso vai nu sarkanā, vai zaļā krāsā (kopumā tieši pusi no visiem cipariem nokrāso sarkanā un otru pusi – zaļā krāsā). No šāda "krāsaina" skaitļa iespējams izveidot divus skaitļus – sarkano un zaļo. Sarkano skaitli veido no sarkanajā krāsā nokrāsotiem cipariem, nemainot to sākotnējo secību. Zaļo skaitli veido analogiski no zaļajā krāsā nokrāsotiem cipariem.

Piemēram, ja dots skaitlis 314159 un tā pirmo, otro un sesto ciparu nokrāso sarkanus, bet pārējos – zaļus, tad sarkanais skaitlis ir 319, bet zaļais skaitlis ir 415. Šo skaitļu absolūtā starpība (rezultāts, ko iegūst no lielākā skaitļa atņemot mazāko) ir $415 - 319 = 96$. Savukārt, ja sarkanā krāsā nokrāso pirmo, trešo un piekto ciparu, tad sarkanais skaitlis ir 345, zaļais skaitlis ir 119 un šo skaitļu absolūtā starpība ir $345 - 119 = 226$.

Uzrakstiet programmu, kas dotam skaitlim noskaidro, kādu **vislielāko** sarkanā un zaļā skaitļa absolūto starpību ir iespējams iegūt, nokrāsojot tā ciparus atbilstoši uzdevuma nosacījumiem!

Ievaddati

Teksta faila `starpiba.dat` pirmajā rindā dots naturāls skaitlis $N(N \leq 300, N - \text{pāra skaitlis})$ – krāsojamā skaitļa ciparu skaits. Faila otrajā rindā dots krāsojamais skaitlis – nenulles ciparu virkne garumā N .

Izvaddati

Teksta faila `starpiba.rez` vienīgajā rindā jāizvada viens vesels nenegatīvs skaitlis – vislielākā iespējamā sarkanā un zaļā skaitļa absolūtā starpība.

Piemēri

Ievaddati (<code>starpiba.dat</code>)	Izvaddati (<code>starpiba.rez</code>)
2 44	0
4 1231	19
6 314159	234