Uzdevuma nosaukums:	**Uzvarētājs**	**Nelegālais dragreiss**	**Stabiņi**
Ievaddatu faila nosaukums:	`uzvara.dat`	`dragreiss.dat`	`stabi.dat`
Izvaddatu faila nosaukums:	`uzvara.rez`	`dragreiss.rez`	`stabi.rez`
Klases vārds risinājumam valodā *Java*	`Uzvara`	`Dragreiss`	`Stabi`
Izpildes laika ierobežojums vienam testpiemēram sekundēs (laiks tiek mērīts uz testēšanas servera):	**Pascal / C / C++**		
	0,4	0,5	0,9
	Java		
	0,9	1,1	1,9

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 ir **100**. Lai risinājums tiktu atzīts par derīgu pamattestēšanai, tam jāizdod pareiza atbilde **visiem** uzdevuma formulējumā dotajiem **piemēriem**.

Uzdevumu tekstos lietotais pieraksts A ≤ x,y,z ≤ B (kur A un B – skaitļi, bet x, y un z – kādi aprakstā lietoti mainīgie), nozīmē, ka vieni un tie paši skaitliskie ierobežojumi attiecas uz katru mainīgo atsevišķi, t.i., vienlaikus ir spēkā sakarības: A ≤ x ≤ B, A ≤ y ≤ B un A ≤ z ≤ B. Līdzīgi, x,y < 100 nozīmē, ka vienlaikus x < 100 un y < 100.

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL:

- FreePascal (versija 2.6.4) ar parametriem
  `-O2 -XS -Sg -Cs64000000`

Valodai C:

- GNU C (versija 4.9.2) ar parametriem
  `-std=gnu99 -O2 -s -static -lm -xc -Wformat -Werror=format`

Valodai C++:

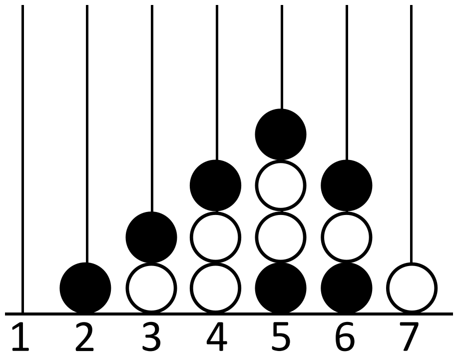
- GNU C++ (versija 4.9.2) ar parametriem
  `-std=gnu++11 -O2 -s -static -xc++ -Wformat -Werror=format`

Valodai Java:

- Java7 (versija OpenJDK 1.7.0_65 jeb 7u65)
  `-Xmx256m -Xms256m -Xss256m`

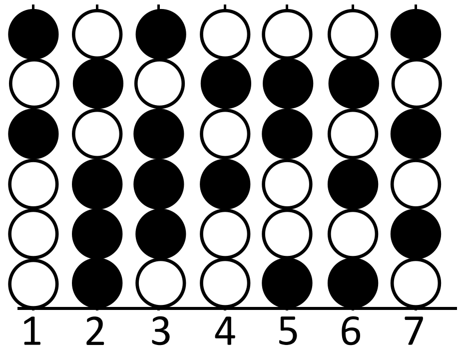
Uzvarētājs

Spēles "DEBESMANNA" piederumi sastāv no vienā plaknē izvietotiem 7 stieņiem, 21 melnas un 21 baltas bumbiņas. Visas bumbiņas ir vienādi lielas un jebkuru no tām var uzvērt uz jebkura no stieņiem. Visi stieņi ir vienādi un uz katra no tiem var būt uzvērtas ne vairāk kā sešas bumbiņas. Stieņi ir numurēti, sākot no kreisās puses pēc kārtas ar naturāliem skaitļiem no 1 līdz 7. Spēli spēlē divi spēlētāji - viens ar baltās krāsas bumbiņām (spēlētājs “B”), otrs ar melnās (spēlētājs “M”). Spēles sākumā visi stieņi ir tukši un spēli sāk spēlētājs “B”. Katrs spēlētājs pēc kārtas uzver vienu savas krāsas bumbiņu uz kāda no stieņiem. Uzvērtā bumbiņa nokrīt maksimāli zemu - līdz pamatam vai līdz pēdējai uz attiecīgā stieņa uzvērtajai bumbiņai. Nav atļauts pārvietot iepriekšējos gājienos uzvērtās bumbiņas. Katra spēlētāja mērķis ir panākt, lai četras no viņa izvietotajām bumbiņām atrastos vienā rindā - horizontāli, vertikāli vai pa diagonāli uz secīgiem stieņiem. Spēlētājs, kuram pirmajam izdodas šo mērķi sasniegt, ir uzvarējis. Ja visas 42 bumbiņas ir uzvērtas un nevienam no spēlētājiem nav izdevies uzvarēt, spēle ir beigusies neizšķirti. Gājieni tiek numurēti pēc kārtas, sākot ar 1 – spēlētājs “B” izdara nepāra, bet “M” – pāra gājienus.



1.zīm. Spēlētāja “M” uzvara.

1.zīmējumā parādītajā situācijā spēlētājs “M” ir uzvarējis 14.gājienā (jo melnās bumbiņas uz blakus esošiem stieņiem 2,3,4 un 5 ir izvietotas rindā pa diagonāli). Gājieni varēja būt izdarīti šādā secībā: 3, 5, 4, 6, 5, 2, 4, 4, 6, 6, 7, 3, 5, 5.



2.zīm. Pilnībā aizpildīti stieņi.

Spēlētāji pēc viena spēlētāja uzvaras turpina izdarīt gājienus pēc kārtas kā iepriekš un pilnībā aizpilda visus stieņus. Ja visi stieņi ir pilnībā aizpildīti, tad abu krāsu bumbiņas ir vienādā skaitā – pa 21. Piemēram, 1.zīmējumā redzamās spēles turpinājums, kad visi stieņi ir pilnībā aizpildīti, parādīts 2.zīmējumā. Protams, ja ir zināms tikai visu stieņu aizpildījums, tad nav vienkārši noteikt, kurš no spēlētājiem un kurā gājienā varēja būt uzvarējis.

Uzrakstiet programmu, kas dotam pilnībā aizpildītam laukumam nosaka, vai spēlē varēja uzvarēt katrs no spēlētājiem un ja varēja, tad kurā gājienā, agrākais, tas varēja notikt!

Ievaddati

Teksta datne **uzvara.dat** satur pilnībā aizpildītu stieņu aprakstu un sastāv no sešām rindām, kur katrā rindā ir septiņi simboli, kas raksturo attiecīgās bumbiņas krāsu. Katram i(1≤i≤6) un j(1≤j≤7) j-tais simbols datnes i-tajā rindā atbilst i-tajai bumbiņai (skaitot no augšas) uz j-tā stieņa. Simbols ‘M’ atbilst melnajai, bet ‘B’ – baltajai bumbiņai. Zināms, ka dotā bumbiņu konfigurācija ir iegūstama iepriekš aprakstītajā veidā dalībniekiem gājienus izdarot pārmaiņus pēc kārtas.

Izvaddati

Teksta datnes **uzvara.rez** pirmajā rindā jāizvada vesels nenegatīvs skaitlis – tā gājiena numurs, kurā spēli ātrākais varēja uzvarēt spēlētājs "B" vai 0, ja spēlētājs "B" šo spēli uzvarēt nevarēja. Datnes otrajā rindā jāizvada vesels nenegatīvs skaitlis – tā gājiena numurs, kurā spēli ātrākais varēja uzvarēt spēlētājs "M" vai 0, ja spēlētājs "M" šo spēli uzvarēt nevarēja.

Piemēri

Ievaddati (uzvara.dat)	Izvaddati (uzvara.rez)	Piezīme
MBMBBBM BMBMMMB MBMBMBM BMMMBMB BMMBBBM BMBBMMB	13 10	Atbilst uzdevuma tekstā dotajam piemēram.

Ievaddati (uzvara.dat)	Izvaddati (uzvara.rez)
BMMBMMB BBBMMMB MMBMBMB BBBMMBM BMMBMBM BMMBBMB	0 0

Apakšuzdevumi un to vērtēšana

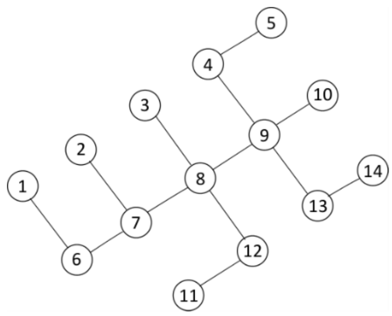
Nr.	Testu apraksts	Punkti
1.	Laukumā četras bumbiņas rindā ir tikai vienam spēlētājam	20
2.	Bez papildus ierobežojumiem	80
	Kopā:	100

Nelegālais dragreiss

Pretstatā dragreisa sacīkstēm, ko rīko oficiāli atzītas organizācijas, ir arī neoficiālas sacensības, kas ir daļa no tā sauktās "ielu sacīkšu kultūras". Lai gan drošības apsvērumu dēļ tās ir aizliegtas, tomēr šādas sacensības notiek. Saskaņā ar http://drag-racing.lv/nelegalais-dragreiss/, "*Nelegālais dragreiss notiek jebkurā vietā, kur bez lieliem traucēkļiem iespējams mērot taisnu distanci. Parasti šis "trases" posms nav noteikta garuma un brauciens notiek līdz kādam noteiktam punktam vai zīmīgai līnijai. Parasti tiek izvēlēti taisni ceļa posmi un sacensības noris diennakts tumšajā laikā, kad ir daudz mazāka satiksmes plūsma.*"

Kukšinieku pagasta vadītājs Paulis ir norūpējies, jo jau vairākas reizes pagasta centrā ir bijusi dzirdama mašīnu rūkoņa, kas vedina uz domām, ka notikuši nelegāli dragreisi. Katra Kukšinieku *iela* sākas un beidzas kādā apļveida *krustojumā*. Iespējams arī, ka krustojumā šī ir vienīgā iela un tad šādu krustojumu sauksim par *strupceļa krustojumu*. No katra krustojuma līdz jebkuram citam krustojumam iespējams aizbraukt pa vienu vai vairākām secīgām (tādām, kuru galapunkts atrodas tajā pašā krustojumā) ielām tikai vienā veidā. Paulis ir nolēmis ierobežot nelegālo dragreisistu aktivitātes, izvietojot krustojumos pašvaldības policistus. Ir zināms, ka nelegālo sacīšu dalībnieki nevēlas iedzīvoties nepatikšanās ar policiju, tāpēc nekad nebrauks cauri krustojumam, kurā atrodas policists. Tā kā nav zināms, cik tālu no krustojuma sākas dragreisa trase, uzskatīsim, ka tā var sākties un beigties vai nu strupceļa krustojumā, vai arī krustojumā, kurā atrodas policists.

Diemžēl policistu skaits ir ierobežots, tāpēc Pauļa uzdevums ir izvietot policistus krustojumos tā, lai lielākais secīgi izbraucamo ielu skaits būtu pēc iespējami mazāks. Katrā krustojumā drīkst atrasties ne vairāk kā viens policists.



Piemēram, ja Pauļa rīcībā ir divi policisti un Kukšinieku ielu un krustojumu tīkls ir tāds, kā redzams zīmējumā (krustojumi numurēti ar naturāliem skaitļiem no 1 līdz 14 pēc kārtas), tad policisti jāizvieto, piemēram, 7. un 9. krustojumā, tādejādi ierobežojot garāko iespējamo nelegālā dragreisa trasi līdz trim secīgām ielām (3→8→12→11, 7→8→12→11, 9→8→12→11 vai šie paši maršruti pretējā secībā). Tādu pašu rezultātu – ne vairāk kā trīs secīgas ielas, var iegūt, ja policistus izvieto 8. un 9. krustojumā – tad iespējamā garākā trase ir 1→6→7→2 (vai 2→7→6→1).

Uzrakstiet programmu, kas atrod tādu policistu izvietojumu, lai garākās iespējamās nelegālā dragreisa trases garums būtu īsākais iespējamais!

Ievaddati

Teksta datnes **dragreiss.dat** pirmajā rindā dotas divu naturālu skaitļu N (krustojumu skaits, (N ≤ 100000) un P (policistu skaits, 1 ≤ P ≤ N) vērtības, kas atdalītas ar tukšumzīmi.

Krustojumi ir numurēti ar naturāliem skaitļiem no 1 līdz N pēc kārtas.

Nākamajās N-1 datnes rindās dots ielu apraksts. Katrā rindā doti divi naturāli skaitļi – ielas sākuma un beigu krustojuma numuri, kas atdalīti ar tukšumzīmi. Katras ielas apraksts ievaddatos dots vienreiz.

Izvaddati

Teksta faila **dragreiss.rez** pirmajā rindā jāizvada naturāls skaitlis - mazākais iespējamais garākās trases garums.

Datnes otrajā rindā jāizvada P dažādi naturāli skaitļi – to krustojumu numuri, kuros jāizvieto policisti. Starp katriem diviem blakus skaitļiem izvaddatos jābūt tukšumzīmei. Krustojumu numurus drīkst izvadīt patvaļīgā secībā.

Piemērs

Ievaddati (dragreiss.dat)	Izvaddati (dragreiss.rez)	Piezīme
14 2 1 6 5 4 6 7 8 7 13 9 2 7 3 8 12 8 8 9 10 9 4 9 11 12 13 14	3 8 9	Atbilst uzdevuma tekstā dotajam piemēram.

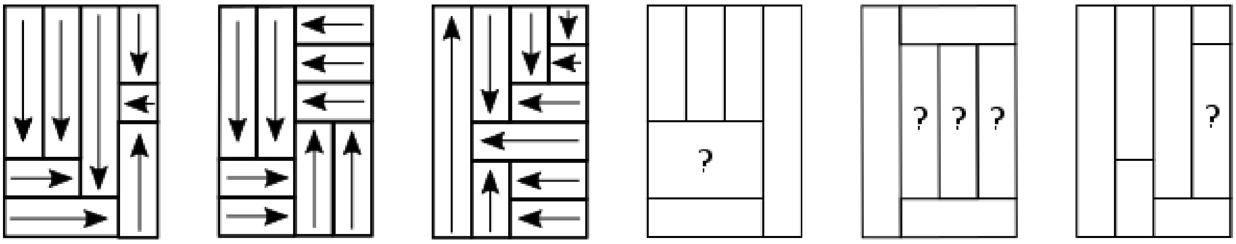
Ievaddati (dragreiss.dat)	Izvaddati (dragreiss.rez)
3 3 1 2 3 2	1 2 3 1

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	N≤100, P=1	5
2.	N≤1000, P=1	10
3.	Bez papildus ierobežojumiem	85
	Kopā:	100

Stabiņi

Teiksim, ka taisnstūrveida N×M rūtiņu laukums ir sadalīts *orientētos stabiņos*, ja katra tā rūtiņa pieder vienu rūtiņu platam taisnstūrim un kāda vienu rūtiņu platā mala ir daļa no kādas laukuma malas. Teiksim, ka šajā vietā atrodas stabiņa *sākums*. Piemēram, zīmējumā a), b) un c) laukumi ir sadalīti orientētos stabiņos (bultas ir vērstas prom no stabiņa sākuma), bet d), e) un f) – nav (neatbilstošie apgabali atzīmēti ar jautājuma zīmēm).



a) b) c) d) e) f)

Uzrakstiet programmu, kas dotām N un M vērtībām nosaka, cik dažādos veidos laukumu var sadalīt orientētos stabiņos! Ievērojiet, ka stabiņi var arī beigties uz laukuma malas un ka līdzīga izskata stabiņi var būt orientēti dažādos veidos!

Ievaddati

Teksta datnes **stabi.dat** pirmajā rindā dotas divu naturālu skaitļu N un M vērtības (N ≤ M ≤ 1000), kas atdalītas ar tukšumzīmi.

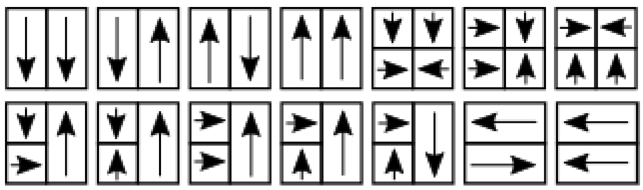
Izvaddati

Teksta faila **stabi.rez** vienīgajā rindā jāizvada naturāls skaitlis – dažādo laukuma sadalīšanas veidu skaits pēc moduļa 10^9+7.

Piemērs

Ievaddati (stabi.dat)	Izvaddati (stabi.rez)
2 2	56

Daži no šiem veidiem ir:



Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	N = 1	10
2.	N = 2	10
3.	N + M ≤ 9	10
4.	M ≤ 200	30
5.	Bez papildus ierobežojumiem	40
	Kopā:	100