Uzdevuma nosaukums:	Zebra	Ekskursija	Vilcieniņš
Ievaddatu datnes nosaukums:	`zebra.dat`	`eks.dat`	`vilc.dat`
Izvaddatu datnes nosaukums:	`zebra.rez`	`eks.rez`	`vilc.rez`
Izpildes laika ierobežojums vienam testpiemēram sekundēs (laiks tiek mērīts uz testēšanas servera):	0,2	0,6	0,3
Atgriezeniskā saite:	Katra apakšuzdevuma vienas grupas testu izpildes rezultāts būs redzams sacensību laikā.		

Ievaddatu un izvaddatu datņu nosaukumi jānorāda **bez** pilnā ceļa (uzskatiet, ka tās atrodas tekošajā katalogā) un tieši tā, kā norādīts uzdevuma formulējumā (**ar mazajiem burtiem**).

Izpildes laika atmiņas ierobežojums: **256 MB**.

Maksimāli iespējamais punktu skaits par uzdevumu: **100**.

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

Viens un tas pats tests vai testu grupa var atbilst vairākiem apakšuzdevumiem. Ir garantēts, ka visi noteikta apakšuzdevuma testi atbilst šī apakšuzdevuma aprakstā dotajiem ierobežojumiem, bet ne tas, ka visi dotā uzdevuma testi ar šādiem ierobežojumiem ir iekļauti šajā apakšuzdevumā.

Kompilējot programmas uz servera, tiks lietoti šādi kompilatori:

Valodai PASCAL:

- FreePascal (versija 2.6.2) ar parametriem
  `-O2 -XS -Sg -Cs64000000`

Valodai C:

- GNU C (versija 4.7.3) ar parametriem
  `-std=gnu99 -O2 -s -static -lm -xc`

Valodai C++:

- GNU C++ (versija 4.7.3) ar parametriem
  `-O2 -s -static -xc++`

Zebra

Uz milzīgas lapas uzzīmēts pelēks kvadrātveida rūtiņu režģis, kur katras rūtiņas malas garums ir viena vienība. Visas režģa rūtiņas izkrāsotas baltā vai melnā krāsā. Krāsojums ir pamīšus pa rindām – ja divām rūtiņām ir kopīga vertikāla mala, tad tās ir nokrāsotas vienā krāsā, bet ja kopīga ir horizontāla mala tad tās ir nokrāsotas dažādās krāsās. Gan vertikālās, gan horizontālās režģa līnijas sanumurētas ar veseliem skaitļiem pēc kārtas. Rūtiņas stūra koordinātas ir attiecīgās vertikālās un horizontālās līnijas numuri. Rūtiņa, kuras stūru koordinātas ir (0,0), (0,1), (1,0) un (1,1), ir izkrāsota baltā krāsā.

Uz rūtiņu laukuma ir uzzīmēts daudzstūris ar N virsotnēm, kura visas virsotnes atrodas rūtiņu režģa krustpunktos. Jānītis vēlas, lai visa daudzstūra iekšpuse būtu nokrāsota melnā krāsā, tāpēc viņam ir svarīgi aprēķināt cik ir baltās daļas laukums daudzstūra iekšpusē (jo tas būs jāpārkrāso), lai varētu nopirkt pietiekami daudz melnās krāsas.

Uzrakstiet programmu, kas dotam daudzstūrim aprēķina baltās daļas laukumu daudzstūrī!

Ievaddati. Teksta datnes **`zebra.dat`** pirmajā rindā dota naturāla skaitļa N vērtība ($3 \le N \le 10^5$). Nākamajās N rindās seko daudzstūra virsotņu apraksts – katrā rindā divi veseli skaitļi, kas atdalīti ar tukšumzīmi, i-tās virsotnes koordinātas – x_i un y_i ($-10^9 \le x_i, y_i \le 10^9$). Virsotnes dotas to apstaigāšanas secībā, kādā tās atrodas uz daudzstūra perimetra.

Izvaddati. Teksta datnes **`zebra.rez`** vienīgajā rindā jāizvada reāls skaitlis – balto rūtiņu kopējais laukums daudzstūra iekšpusē. Atbilde tiks uzskatīta par pareizu, ja tā atšķirsies no pareizās atbildes ne vairāk kā par 10^{-6}.

Piemēri.

Ievaddati `zebra.dat`	Izvaddati `zebra.rez`
3 1 0 1 1 0 2	0.25

Ievaddati `zebra.dat`	Izvaddati `zebra.rez`
5 1 1 1 3 2 3 3 2 3 1	1.5

Apakšuzdevumi un to vērtēšana.

Nr.	Testu apraksts	Punkti
1.	N ≤ 1000	20
2.	Zināms, ka daudzstūris ir izliekts	30
3.	Bez papildus ierobežojumiem	50
	Kopā:	**100**

Ekskursija

Skolotāja Alma skolēniem ir noorganizējusi ekskursiju, kurā paredzēts apskatīt K dažādas pilsētas. Pieteikušos skolēnu skaits pārsniedza viena autobusa ietilpību, tāpēc ekskursijai tika pasūtīti divi autobusi, kur katra autobusa ietilpība ir N pasažieri. Dīvainas sagadīšanās dēļ uz ekskursiju ieradās tieši N skolēni, kas, ieskaitot Almu, joprojām neietilpst vienā autobusā. Lai novērstu skolēnu vēlmi visiem saspiesties vienā autobusā un Almu atstāt vienu pašu otrajā, Alma ir nolēmusi ekskursijā ieviest spēles elementus.

Katrā brīdī visiem skolēniem ir noteikta secība, ko ekskursijas sākumā nosaka tas, kādā secībā skolēni ir ieradušies pulcēšanās vietā. Skolēnu secība var mainīties tikai pēc ierašanās kārtējā pilsētā. Visas ekskursijas laikā notikušās iekāpšanas autobusos reizes sanumurēsim pēc kārtas no $i = 0$ (iekāpšana ekskursijas sākumā) līdz $i = K$ (iekāpšana pēc pēdējās pilsētas apskates). Katrā iekāpšanas reizē (gan ekskursijas sākumā, gan pēc katras pilsētas apmeklējuma) pirmajā autobusā iekāpj pirmie x_i skolēni, bet pārējie – otrajā, kur x_i ($1 \le x_i < N$) – Almas izdomāts skaitlis tieši šai iekāpšanas reizei. Pie tam skolēni vietas autobusā ieņem pēc principa „pirmais iekšā – pirmais ārā" – t.i., skolēni no katra autobusa izkāps tieši tādā pat secībā, kādā tajā iekāpuši.

Skolēnu secību i-tās pilsētas apskatīšanai nosaka šādi: Alma salīdzina skolēnu, kuriem ir kārta izkāpt no katra no autobusiem un pirmais izkāpj pēc auguma īsākais no tiem. Ja abi skolēni ir vienādi gari, tad priekšroka ir skolēnam no pirmā autobusa. Kad kāds skolēns izkāpj no autobusa, izkāpt no šī autobusa gatavojas nākamais skolēns pēc kārtas un atkal tiek salīdzināts ar skolēnu no otra autobusa. Tā šis process turpinās, līdz visi skolēni ir izkāpuši. Šāda secība saglabājas visu pilsētas apskates laiku un beidzas atkal ar kārtējo iekāpšanu autobusos. Uzrakstiet programmu, kas zināmai skolēnu secībai, kādā tie ieradās uz ekskursiju, skolēnu augumiem un Almas izvēlētajiem skaitļiem nosaka kāda būs skolēnu secība pēdējās (K-tās) pilsētas apskates laikā.

Ievaddati. Teksta datnes **`eks.dat`** pirmajā rindā dotas naturālu skaitļu N (skolēnu skaits, $1 < N \le 10^5$) un K (apskatāmo pilsētu skaits, $K \le 10^5$) vērtības, kas atdalītas ar tukšumzīmi. Otrajā datnes rindā doti skolēnu augumi – N naturāli skaitļi, kas atdalīti ar tukšumzīmēm. Neviena skolēna augums nepārsniedz 2×10^9 nanometrus. Datnes trešajā rindā doti K naturāli skaitļi, kas atdalīti ar tukšumzīmēm. Katram i ($0 \le i < K$) i+1-ais skaitlis rindā norāda skolēnu skaits, kas i-tajā iekāpšanas reizē iekāps pirmajā autobusā. Tā kā uzdevumā nepieciešams noteikt skolēnu secību K-tās pilsētas apskates laikā, tad nav nepieciešams zināt skolēnu skaitu, kas pirmajā autobusā iekāps pēdējā iekāpšanas reizē (pēc K-tās pilsētas apskates) un šis skaitlis ievaddatos nav dots.

Izvaddati. Teksta datnes **`eks.rez`** vienīgajā rindā jāizvada N naturāli skaitļi – skolēnu augumu secība K-tās pilsētas apskates laikā.

Piemērs.

Ievaddati **`eks.dat`**	Izvaddati **`eks.rez`**	Piezīme *(skolēnu secība katras pilsētas apskates laikā)*

`8 3`
`6 1 9 5 2 8 6 13`
`3 5 7`
	`5 2 6 1 6 8 9 13`	1.: **5 2 6 1 8 6 9 13**
		2.: **5 2 6 1 6 8 9 13**
		3.: **5 2 6 1 6 8 9 13**

Apakšuzdevumi un to vērtēšana.

Nr.	Testu apraksts	Punkti
1.	Visu skolēnu augumi ir atšķirīgi	30
2.	$N \times K \le 10^6$	5
3.	Visas x_i vērtības ir vienādas	10
4.	Bez papildus ierobežojumiem	55
	Kopā:	**100**

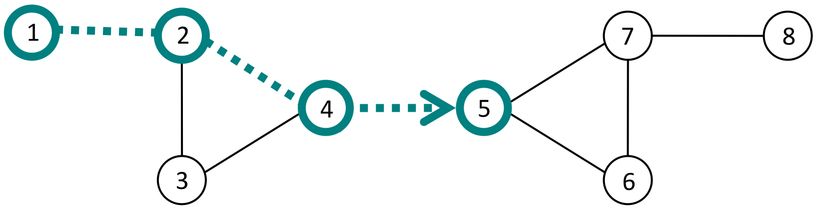
Vilcieniņš

Atrakciju parka V organizētāju plānos ir piedāvāt braucienus ar vilcieniņu, kas pārvietojas izmantojot atrakciju parka sliežu tīklu. Vilcieniņš sastāv no k virknē savienotiem vagoniem, kur pirmais vagons ir arī lokomotīve. Atrakcijā pavisam ir n pieturas, sanumurētas ar skaitļiem no 1 līdz n, katrā no kurām var atrasties vilcieniņa vagons. Dažus pieturu pārus savā starpā savieno viens sliežu ceļš: ja pietura a ir savienota ar b, tad arī b ir savienota ar a. Pavisam atrakciju parkā savienoti ir m pieturu pāri.

Pieturas ir izvietotas tik tuvu, ka divi secīgi vilcieniņa vagoni vienmēr atrodas divās blakus pieturās, kuras tieši savieno sliežu ceļš. Sākumā vilcieniņa vagoni atrodas pieturās v_1, v_2, …, v_k, pie tam pieturā v_1 atrodas vilcieniņa lokomotīve. Nevienā pieturā vienlaikus nevar atrasties divi vagoni: visi v_i ir dažādi. Katram i ($1 \le i < k$), pieturas v_i un v_{i+1} tieši savieno sliežu ceļš. Vilcieniņš var kustēties pa sliedēm tā, ka lokomotīvei pārvietojoties uz pieturu, kas tieši savienota ar pašreizējo lokomotīves pieturu, visi pārējie vagoni pārvietojas līdzi lokomotīvei vienu pieturu uz priekšu. Formāli: ja eksistē tāda pietura u, kas ir vai nu brīva, vai arī tajā atrodas vilcieniņa pēdējais vagons v_k, un u ir savienota ar v_1, tad lokomotīve var pārvietoties uz to: $v_1' = u$, un visi vagoni pārvietojas tā, ka i-tā vagona jaunā atrašanās vieta ir pietura $v_i' = v_{i-1}$ (visiem i, $1 < i \le k$).

Atrakciju parka sliežu tīkls ir uzbūvēts īpašā veidā. Par *ciklu* sauksim tādu pieturu virkni a_1, a_2, …, a_s, kur $s \ge 3$, ka visiem i ($1 \le i < s$), pieturu pārus a_i un a_{i+1}, kā arī a_s un a_1, tieši savieno sliežu ceļš. Sliežu tīklam ir spēkā apgalvojums: katra pietura atrodas ne vairāk kā vienā ciklā.

Vilcieniņš jau atrodas savā sākumpozīcijā. Tagad nepieciešams noskaidrot, kurām pieturām u eksistē tāds vilcieniņa ceļojums, ka pārvietošanās beigās lokomotīve nokļūst virsotnē u. Dotajam sliežu tīkla aprakstam, vilcieniņa garumam un tā vagonu sākumpozīcijai tīklā nosakiet visas iespējamās lokomotīves galapieturas!



Piemērā ir attēlots sliežu tīkls un vilcieniņa sākumstāvoklis (lokomotīve atzīmēta ar bultiņu). Šajā gadījumā vilcieniņš var vai nu nekustēties (lokomotīve paliek pieturā ar numuru 5), vai arī pārvietot lokomotīvi uz vienu no pieturām ar numuriem 6 un 7. No pieturas ar numuru 7 lokomotīvi varēs pārvietot uz pieturu ar numuru 8. Uz pārējām pieturām no dotā sākumstāvokļa vilcieniņa lokomotīve nokļūt nevar.

Ievaddati. Teksta datnes **`vilc.dat`** pirmajā rindā doti divi veseli skaitļi n un m ($1 \le n, m \le 10^5$), kas atdalīti ar tukšumzīmi – pieturu skaits un savienoto pieturu pāru skaits sliežu tīklā. Nākamajās m rindās ir dots sliežu ceļu apraksts: i-tajā datnes rindā doti divi veseli skaitļi a_i un b_i ($1 \le a_i, b_i \le n$; $a_i \ne b_i$), kas atdalīti ar tukšumzīmi – i-tā savienoto pieturu pāra numuri. Katrs savienoto pieturu pāris ievaddatos dots tieši vienreiz. Garantēts, ka sliežu tīkls ir saistīts, t.i., jebkuras divas dažādas pieturas ir tieši vai netieši savienotas savā starpā. Katra pietura atrodas ne vairāk kā vienā ciklā. Nākamajā datnes rindā ir dots vesels skaitlis k ($3 \le k \le n$), vilcieniņa garums. Pēdējā datnes rindā ir doti k veseli skaitļi v_1, v_2, …, v_k ($1 \le v_i \le n$; $v_i \ne v_j$, ja $i \ne j$), vilcieniņa sākumstāvoklis. i-tais vagons, sākot ar lokomotīvi, pēc kārtas atrodas pieturā v_i. Garantēts, ka katram i ($1 \le i < k$) pieturas v_i un v_{i+1} ir tieši savienotas ar sliežu ceļu.

Izvaddati. Teksta datnes **`vilc.rez`** pirmajā rindā jāizvada vesels skaitlis *p* – iespējamo lokomotīves gala pieturu skaits tīklā. Nākamajā rindā ir jāizvada *p* veseli skaitļi augošā secībā – visu iespējamo lokomotīves gala pieturu numuri.

Piemērs (atbilst tekstā dotajam zīmējumam).

Ievaddati **`vilc.dat`**	Izvaddati **`vilc.rez`**
8 9 1 2 2 3 3 4 4 2 4 5 5 6 6 7 7 5 7 8 4 5 4 2 1	4 5 6 7 8

Apakšuzdevumi un to vērtēšana.

Nr.	Testu apraksts	Punkti
1.	Sliežu tīklā nav neviena cikla	10
2.	Sliežu tīklā ir tieši viens cikls	20
3.	Bez papildus ierobežojumiem	70
	Kopā:	**100**