

**LATVIJAS 24. INFORMĀTIKAS OLIMPIĀDES III POSMS**  
**JAUNĀKĀS GRUPAS UZDEVUMU APSKATS**  
**Pirmā diena (2011.gada 22.marts)**



Uzdevuma nosaukums:	Kino	Skaitļi un cipari	Lielpilsētas siluets
Ievaddatu faila nosaukums:	kino.dat	cipari.dat	siluets.dat
Izvaddatu faila nosaukums:	kino.res	cipari.res	siluets.res
Izpildes laika ierobežojums vienam testpiemēram (laiks tiek mērīts uz testēšanas servera):	0,3 sekundes	0,2 sekundes	0,5 sekundes
Atmiņas ierobežojums:	64MB	64MB	64MB
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`

Valodai C:

- GNU C (versija 3.4.2 un 4.4.1) ar parametriem `-std=c99 -O2 -s -static -lm`
- Microsoft Visual C 2008 ar parametriem `/TC /O2`

Valodai C++:

- GNU C++ (versija 3.4.2 un 4.4.1) ar parametriem `-O2 -s -static`
- Microsoft Visual C++ 2008 ar parametriem `/TP /O2`

**LATVIJAS 24. INFORMĀTIKAS OLIMPIĀDES III POSMS**  
**JAUNĀKĀS GRUPAS UZDEVUMI**  
**Pirmā diena (2011.gada 22.marts)**

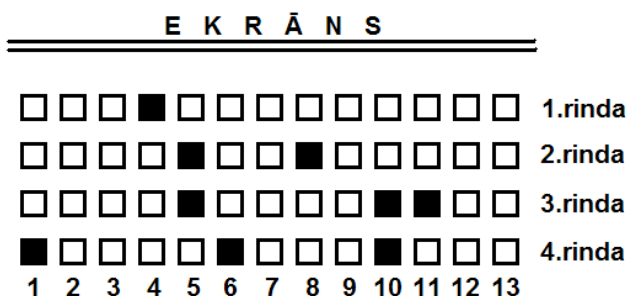


**Kino**

Skolotāja Alma nolēmusi ar klasi beidzot apmeklēt jauno filmu „Kaprāja Liepiņa aizbraukšana”. Filmu demonstrē kinoteātrī „E-Laime”, kurā ir N krēslu rindas pa M krēsliem katrā. Alma šajā kinoteātrī jau ir bijusi un zina, ka uz filmas seansu var būt labas un ne tik labas vietas. Vieta ir laba, ja vienlaikus:

- tā neatrodas pārāk tuvu ekrānam (rindas numurs ir vismaz  $R_{MAZ}$ );
- biļete uz šo vietu līdz šim nav pārdota;
- nav pārdota biļete arī uz vietu ar šādu pašu vietas numuru iepriekšējā rindā (neattiecas uz pirmās rindas vietām).

Piemēram, ja  $N=4$ ,  $M=13$ ,  $R_{MAZ}=2$  un līdz šim pārdotas deviņas biļetes (skat. zīm.), tad pirmajā rindā labu vietu nav (par tuvu), otrajā rindā labas ir vietas ar numuriem 1,2,3,6,7,9,10,11,12 un 13, trešajā – 1,2,3,4,6,7,9,12 un 13, bet ceturtajā – 2,3,4,7,8,9,12 un 13.



Uzrakstiet programmu, kas dotam kinozāles

un līdz šim pārdoto biļešu aprakstam nosaka, kāds šajā brīdī ir labo vietu skaits!

**Ievaddati**

Teksta faila **kino.dat** pirmajā rindā dotas četrus veselus nenegatīvus skaitļus  $N$  (rindu skaits,  $1 \leq N \leq 10^6$ ),  $M$  (vietu skaits katrā rindā,  $1 \leq M \leq 10^6$ ),  $R_{MAZ}$  (pirmās „labās” rindas numurs,  $1 \leq R_{MAZ} \leq N$ ) un  $P$  (pārdoto biļešu skaits,  $0 \leq P \leq NM$ ,  $P \leq 10^5$ ) vērtības, kas atdalītas ar tukšumzīmēm. Katrā no nākošajām  $P$  faila rindām katrā dots vienas pārdotās biļetes apraksts – divu naturālu skaitļu  $p_r$  (rindas numurs,  $1 \leq p_r \leq N$ ) un  $p_v$  (vietas numurs,  $1 \leq p_v \leq M$ ) vērtības, kas atdalītas ar tukšumzīmi. Informācija par pārdotajām biļetēm failā ir dota rindu numuru pieaugšanas secībā, bet vienai rindai – vietu numuru pieaugšanas secībā.

**Izvaddati**

Teksta faila **kino.rez** vienīgajā rindā jāizvada vesels nenegatīvs skaitlis – labo vietu skaits.

**Piemērs** (atbilst tekstā dotajam piemēram)

Ievaddati (kino.dat)	Izvaddati(kino.rez)
4 13 2 9	27
1 4	
2 5	
2 8	
3 5	
3 10	
3 11	
4 1	
4 6	
4 10	

**LATVIJAS 24. INFORMĀTIKAS OLIMPIĀDES III POSMS**  
**JAUNĀKĀS GRUPAS UZDEVUMI**  
**Pirmā diena (2011.gada 22.marts)**



**Skaitļi un cipari**

Bezgalīgi liela rūtiņu lapa tiek aizpildīta pēc kārtas ar naturāliem skaitļiem sākot no 1 pēc šāda parauga:

Vispirms kādā rūtiņā tiek ierakstīts 1, rūtiņā virs tā – 2, tālāk pa spirāli pulksteņrādītāja virzienā 3,4,5,6,7,8 un 9. Vairākciparu skaitļu cipari tiek rakstīti pēc kārtas katrs savā rūtiņā. Zīmējumā attēlots lapas rūtiņu aizpildījums brīdī, kad ierakstīti skaitļi līdz 19 ieskaitot (pelēkās svītriņas parāda aizpildīšanas secību). Turpinot aizpildīt lapu, kādā brīdī tajā varēs atrast 2x2 rūtiņu fragmentu ar jebkuru iespējamu ciparu aizpildījumu.

9	1	8	1	9	2	0
2	7	1	0	1	1	2
8	1	9	2	3	1	1
2	6	8	1	4	2	2
7	1	7	6	5	1	2
2	5	1	4	1	3	2
6	2	5	2	4	2	3

Piemēram, tagad lapā var atrast fragmentu

2	7
8	1

ko nebija iespējams atrast brīdī, kad lapā bija ierakstīti tikai skaitļi no 1 līdz 28.

Uzrakstiet programmu, kas dotam 2x2 rūtiņu fragmentam nosaka, līdz kādam mazākajam skaitlim N (ieskaitot) jāaizpilda lapa, lai tajā būtu iespējams atrast norādīto fragmentu!

**Ievaddati**

Teksta faila **cipari.dat** pirmajā rindā doti divi cipari, kas atdalīti ar tukšumzīmi – fragmenta augšējā rindā esošie cipari. Faila otrajā rindā doti fragmenta otrajā rindā esošie cipari tādā pat formātā.

**Izvaddati**

Teksta faila **cipari.rez** vienīgajā rindā jāizvada mazākais naturālais skaitlis N, līdz kuram aizpildot lapu, tajā varēs atrast ievaddatos doto fragmentu. Fragmentam lapā jābūt atrodamam tieši tā, kā dots ievaddatos (tas nedrīkst būt pagriezts vai atspoguļots)

**Piemēri**

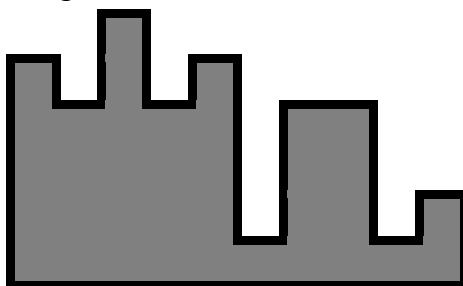
Ievaddati (cipari.dat)	Izvaddati(cipari.rez)
2 7 8 1	29

Ievaddati (cipari.dat)	Izvaddati(cipari.rez)
2 3 1 4	4

Ievaddati (cipari.dat)	Izvaddati(cipari.rez)
0 3 1 8	31

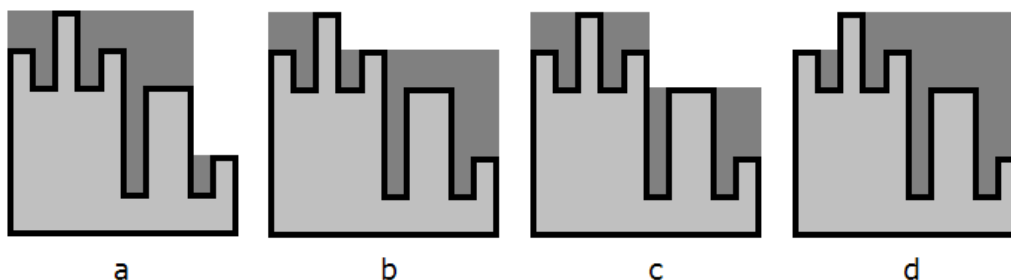
## Lielpilsētas siluets

Lielpilsētas siluets ir dažāda augstuma ēku virkne. Piemēram, tas var būt šāds:



Katrai siluetu veidojošai ēkai ir noteikts skaits vienāda augstuma stāvu un platumā siluetu var vienmērīgi sadalīt vienāda platuma posmos. Katrā posmā ir redzama kāda ēka ar vismaz vienu un ne vairāk kā  $10^6$  stāviem. Pilsētas siluetu var raksturot kā augstākās redzamās ēkas stāvu skaitu katrā no posmiem, pārskaitot posmus pēc kārtas no kreisās puses uz labo. Dotajā piemērā šī virkne būtu 5,4,6,4,5,1,4,4,1,2.

Lielpilsētas mērs ir nolēmis, ka pilsētas siluets izskatītos glītāk, ja to veidotu tieši divu dažādu augstumu ēkas. Pie kam, skatoties no kreisās puses, vispirms pēc kārtas būtu redzamas tikai pirmā augstuma ēkas, bet pēc tam tikai otrā augstuma ēkas. Lai doto siluetu pārveidotu, vienā vai vairākos posmos varētu būt nepieciešams piebūvēt vienu vai vairākus stāvus.



Iepriekš dotajā piemērā iespējami vairāki pilsētas silueta pārveidošanas varianti:

Tiesa, dažādiem variantiem posmos papildus piebūvējamo stāvu kopskaits ir atšķirīgs: 16(a), 17(b), 14(c) un 22(d).

Uzrakstiet programmu, kas nosaka, kāds mazākais stāvu kopskaits posmos jāpiebūvē papildus, lai realizētu lielpilsētas mēra ieceri!

### **levaddati**

Teksta faila **siluets.dat** pirmajā rindā dota naturāla skaitļa  $N(1 < N \leq 500000)$  vērtība. Katrā no nākošajām  $N$  faila rindām dots viens naturāls skaitlis – stāvu skaits augstākajā posmā redzamajā ēkā. Katram  $i(1 \leq i \leq N)$  i-tajā posmā augstākās redzamās ēkas stāvu skaits dots faila  $i+1$ -ajā rindā.

### **Izvaddati**

Teksta faila **siluets.rez** vienīgajā rindā jāizvada vesels nenegatīvs skaitlis – mazākais posmos piebūvējamo stāvu kopskaits.

**LATVIJAS 24. INFORMĀTIKAS OLIMPIĀDES III POSMS  
JAUNĀKĀS GRUPAS UZDEVUMI  
Pirmā diena (2011.gada 22.marts)**



**Piemēri**

levaddati (siluets.dat)	Izvaddati(siluets.rez)	Piezīme
<b>10</b> 5 4 6 4 5 1 4 4 1 2	<b>14</b>	Piemērs atbilst formulējumā dotajam. Mazākais piebūvējamo stāvu kopskaits ir parādīts variantā c

levaddati (siluets.dat)	Izvaddati(siluets.rez)	Piezīme
<b>5</b> 1 1 3 3 3	<b>0</b>	Siluets jau atbilst prasībām – tajā ir tieši divu dažādu augstumu ēkas derīgā secībā (vispirms viena, tad otra augstuma ēkas).