



Uzdevuma nosaukums:	Skaitļu pārveidošana	Komplekti	Gaisma tavos logos
Ievaddatu faila nosaukums:	<code>parveido.dat</code>	<code>komplekti.dat</code>	<code>gaisma.dat</code>
Izvaddatu faila nosaukums:	<code>parveido.rez</code>	<code>komplekti.rez</code>	<code>gaisma.rez</code>
Klases vārds risinājumam valodā <i>Java</i> :	Parveido	Komplekti	Gaisma
Izpildes laika ierobežojums vienam testpiemēram sekundēs (laiks tiek mērīts uz testēšanas servera):	Pascal / C / C++		
	0,1	0,1	0,2
	Java		
	0,3	0,3	0,5

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 $A \leq x, y, z \leq B$ (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: $A \leq x \leq B$, $A \leq y \leq B$ un $A \leq z \leq B$. Līdzīgi, $x, y < 100$ nozīmē, ka vienlaikus $x < 100$ un $y < 100$.

Valodā Java iesūtītajos risinājumos jābūt **tieši vienai publiskai klasei ar norādīto vārdu**. Pārējām klasēm, ja tādas ir, jābūt iekšējām.

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 4.8.1) ar parametriem
`-std=gnu99 -O2 -s -static -lm -xc -Wformat -Werror=format`

Valodai C++:

- GNU C++ (versija 4.8.1) ar parametriem
`-O2 -s -static -xc++ -Wformat -Werror=format`

Valodai Java:

- Java7 (versija OpenJDK 1.7.0_65 jeb 7u65) ar parametriem
`-Xmx256m -Xms256m -Xss64m`



Skaitļu pārveidošana

Aplūkosim tādu naturālu skaitļu pārveidošanas spēli, kuras katrā *gājienā* skaitli drīkst vai nu

- palielināt par 10^n (n – vesels nenegatīvs skaitlis, atcerieties, ka $10^0=1$), vai
- aizstāt kādu ciparu tā decimālajā pierakstā pret kādu citu (pirmo ciparu nedrīkst pārveidot par 0).

Interesēsimies par mazāko iespējamo gājienu skaitu, lai no viena dota skaitļa A iegūtu citu, lielāku, dotu skaitli B.

Piemēram, no skaitļa 31 skaitli 107 var iegūt ātrākais trīs gājienos: $31 \rightarrow 131 \rightarrow 101 \rightarrow 107$, no skaitļa 2909 skaitli 3000 – divos: $2909 \rightarrow 2999 \rightarrow 3000$.

Uzrakstiet programmu, kas dotiem skaitļiem A un B ($B>A$) nosaka, ar kādu mazāko gājienu skaitu no A var iegūt B!

Ievaddati

Teksta faila **parveido.dat** pirmajā rindā dota naturāla skaitļa A, bet otrajā – naturāla skaitļa $B(A<B<10^{100000})$ vērtība.

Izvaddati

Teksta faila **parveido.rez** vienīgajā rindā jāizvada naturāls skaitlis - mazākais nepieciešamais gājienu skaits, lai no skaitļa A iepriekšaprakstītajā veidā iegūtu skaitli B.

Piemēri

Ievaddati (parveido.dat)	Izvaddati (parveido.rez)	Gājieni
3909 7001	3	3909→7909→7009→7001
Ievaddati (parveido.dat)	Izvaddati (parveido.rez)	Gājieni
398 2015	5	398→1398→2398→ 2098→2018→2015

1.apakšuzdevuma testu ievaddati

Ievaddati (parveido.dat)	Ievaddati (parveido.dat)
51239	499999999
56300	500009999

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie divi testi	2
2.	$B \leq 10^5$	8
3.	$B \leq 10^{500}$	30
4.	Bez papildus ierobežojumiem	60
Kopā:		100



Komplekti

Suvenīru ražošanas firma „Niekivien” ražo divu veidu koka suvenīrus – svirpulniekus un pukstiņus un to ražošanai izmanto standarta kokmateriālus – L centimetru garas brusas.

Viena svirpulnieka izgatavošanai nepieciešama LS centimetrus, bet pukstiņa – LP centimetrus gara sagatave, kas nozāģēta no vienas brusas. Firma suvenīrus pārdod komplektos un ir izpētīts, ka pircējiem vislabāk patīk, ja suvenīru komplektā ir tieši NS svirpulnieki un NP pukstiņi.

Firmas vadība vēlas noskaidrot, kādu lielāko sagatavoto suvenīru komplektu skaita attiecību pret izmantoto brusu skaitu iespējams sasniegt un kā šajā gadījumā jāorganizē ražošana.

Pieņemsim, ka brusas garums ir 20 cm, svirpulnieka izgatavošanai nepieciešama 3 cm gara sagatave, pukstiņa izgatavošanai nepieciešama 5 cm gara sagatave un komplektā ietilpst viens svirpulnieks un trīs pukstiņi.

Tad no katras brusas var izgatavot dažādu katra veida suvenīru skaitu. Tabulā uzrādīti tikai tie varianti, kur brusa izmantota maksimāli – no atlikuma (kas arī norādīts tabulā) vairāk nevienu suvenīru izgatavot nevar:

Veids	Svirpulnieku sk.	Pukstiņu sk.	Atlikums(cm)
A	6	0	2
B	5	1	0
C	3	2	1
D	1	3	2
E	0	4	0

Ja visas brusas sazāģē D veidā, tad no katras brusas var izveidot tieši vienu komplektu un attiecība ir 1(komplekts):1(brusa). Savukārt, ja divas brusas sazāģē B veidā, bet septiņas – E veidā, tad minētā attiecība ir 10:9, kas ir ne vien lielāka par iepriekšējo, bet lielākā iespējamā.

Uzrakstiet programmu, kas atrod iespējami labāko komplektu un brusu attiecību un atrod šai attiecībai atbilstošu brusu sazāģēšanas variantu!

ievaddati

Teksta faila **komplekti.dat** pirmajā rindā dots naturāls skaitlis L(brusas garums centimetros, $L \leq 10^6$). Faila otrajā rindā doti divi naturāli skaitļi, kas atdalīti ar tukšumzīmi – LS(viena svirpulnieka izgatavošanai nepieciešamās sagataves garums centimetros, $LS \leq L$) un NS(svirpulnieku skaits komplektā, $NS \leq 10^5$). Faila trešajā rindā doti divi naturāli skaitļi, kas atdalīti ar tukšumzīmi – LP(viena pukstiņa izgatavošanai nepieciešamās sagataves garums centimetros, $LP \leq L$) un NP(pukstiņu skaits komplektā, $NP \leq 10^5$).



Izvaddati

Viens no komplektu izveidošanas variantiem ar lielāko iespējamo izveidoto komplektu un izmantoto brusu skaita attiecību jāapraksta teksta failā **komplekti rez**. Faila pirmajā rindā jāizvada divi naturāli skaitļi, kas atdalīti ar tukšumzīmi – izveidoto komplektu skaits un izmantoto brusu skaits. Izmantoto brusu skaits nedrīkst pārsniegt 10^{12} – visiem testiem ir atrisinājums, kas atbilst šim ierobežojumam.

Faila otrajā rindā jāizvada naturāls skaitlis M – dažādo izmantoto brusu sazāgēšanas veidu skaits.

Katrā no nākamajām M rindām jāizvada informācija par vienu sazāgēšanas veidu un katrai no šīm rindām jāsaturs trīs veseli nenegatīvi skaitļi: cik brusas šādi sazāgētas, cik svirpulnieku un cik pukstiņu iegūst no vienas šādā veidā sazāgētas brusas. Starp katriem diviem blakus skaitļiem izvaddatos jābūt tukšumzīmei.

Piemēri

Ievaddati (komplekti.dat)	Izvaddati (komplekti.res)	Piezīme
20	10 9	Atbilst uzdevuma tekstā aprakstītajam piemēram.
3 1	2	
5 3	2 5 1	
	7 0 4	

Ievaddati (komplekti.dat)	Izvaddati (komplekti.res)
15	1 2
14 1	2
13 1	1 0 1
	1 1 0

1.apakšuzdevuma testu ievaddati

Ievaddati (komplekti.dat)
33
4 2
5 3

Ievaddati (komplekti.dat)
61
11 1
3 3

Ievaddati (komplekti.dat)
99
18 3
4 1

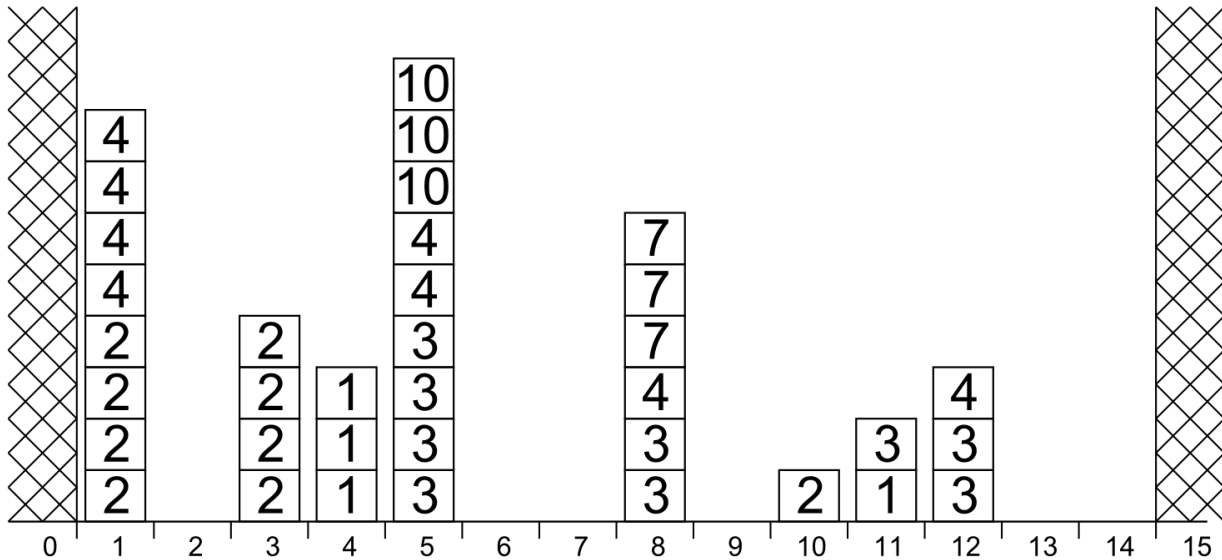
Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie trīs testi	2
2.	$L \leq 100$	8
3.	$L \leq 10000$	15
4.	Bez papildus ierobežojumiem	75
	Kopā:	100



Gaisma tavos logos

Arhitektu birojam „Tasja & Esabruks” ir uzdots projektēt daudzstāvu viesnīcu kompleksu slavenā piejūras kūrortā. Viesnīcu būvei ir atvēlēta vien šaura zemes strēle N vienību garumā, kas stiepjās gar jūru un ko abos galos ierobežo stāvas klintis, pie kam katrā vienībā drīkst būt ne vairāk kā vienu viesnīcu. Schematiski katru viesnīcu var aprakstīt kā naturāla augstuma stabiņus uz veselo skaitļu ass, uz kuras atzīmētas N vienības, kas sanumurētas pēc kārtas ar skaitļiem no 1 līdz N :



Tā kā paredzētais viesnīcu blīvums ir visai liels, ir svarīgi noteikt, cik daudz gaismas nokļūs katras viesnīcas stāva logos. Visu viesnīcu stāvi ir vienādi augsti un katram no tiem tiek aprēķināts *labums*, kas tiek izteikts iepriekšminētajās garuma vienībās. Ja viesnīcu paredzēts būt vienībā i ($1 \leq i \leq N$), tad tās j -tā ($1 \leq j$) stāva labumu aprēķina vispirms uz katru pusi atrodot tuvāko objektu, kas ir vismaz j stāvus augsts (pietiekami augsta viesnīca vai klints) un izvēloties lielāko no šiem attālumiem. Aprēķinos tiek uzskatīts, ka klintis atrodas vienībās 0 un $N+1$, pie kam tās ir augstākas par visaugstāko iespējamo viesnīcu. Zīmējumā katrā rūtiņā ierakstītais skaitlis ir attiecīgā viesnīcas stāva labums.

Uzrakstiet programmu, kas plānotam viesnīcu izvietojumam un katras viesnīcas stāvu skaitam, aprēķina katras viesnīcas visu stāvu kopējo labumu!

Ievaddati

Teksta faila **gaisma.dat** pirmajā dots naturāls skaitlis N ($N \leq 10^5$, zemes strēles garums garuma vienībās). Faila otrajā rindā doti N veseli nenegatīvi skaitļi. Katram i ($1 \leq i \leq N$) i -tais skaitlis šajā rindā apzīmē tās viesnīcas, ko paredzēts būt i -tajā vienībā, stāvu skaitu. 0 nozīmē, ka viesnīcu šajā vienībā nebūvēts. Nevienu viesnīcas augstums nepārsniedz 10^9 . Starp katriem diviem blakus skaitļiem ievaddatos ir viena tukšumzīme.



Izvaddati

Teksta faila **gaisma.rez** pirmajā rindā jāizvada tik skaitļi, cik viesnīcu ir aprakstīts ievaddatos. Šiem skaitļiem jāapraksta viesnīcu labumi pēc kārtas vienību numuru pieaugšanas secībā. Starp katriem diviem blakus skaitļiem izvaddatos jābūt tukšumzīmei.

Piemērs (atbilst uzdevuma tekstā dotajam attēlam)

ievaddati (gaisma.dat)	Izvaddati (gaisma.rez)
14	24 8 3 50 31 2 4 10
8 0 4 3 9 0 0 6 0 1 2 3 0 0	

1.apakšuzdevuma testu ievaddati

ievaddati (gaisma.dat)
21
9 3 6 3 8 3 6 3 7 3 6 3 7 3 6 3 8 3 6 3 9

ievaddati (gaisma.dat)
6
345678912 123456789 456789123 234567891 678912345 567891234

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie divi testi	2
2.	$N \leq 100$, nevienas viesnīcas stāvu skaits nepārsniedz 100	8
3.	$N \leq 1000$	20
4.	Bez papildus ierobežojumiem	70
Kopā:		100