

LATVIJAS 21. INFORMĀTIKAS OLIMPIĀDES
I POSMA UZDEVUMI
VECĀKAJAI (11.-12. KLAŠU) GRUPAI



1. “NK-VIRKNES GARUMS”

(100 punkti)*

Par *NK-virkni* saucim ciparu virkni, ko iegūst, vispirms uzrakstot naturālu skaitli N , tam galā pierakstot skaitli N^*2 , pēc tam N^*3 , ..., N^*K (kur K – naturāls skaitlis). Piemēram, ja $N = 6$ un $K = 5$, iegūsim *NK-virkni* 612182430.

Uzrakstiet programmu, kas dotām N un K vērtībām atrod *NK-virknes garumu* (tajā ietilpstošo ciparu skaitu)!

Ievaddati

Teksta faila `nk.dat` pirmajā rindā doti divi naturāli skaitļi N un K , kas atdalīti ar tukšumzīmi. Zināms, ka N^*K vērtība nepārsniedz $2*10^9$.

Izvaddati

Teksta faila `nk.rez` vienīgajā rindā jāizvada naturāls skaitlis – *NK-virknes garums*. Zināms, ka nekādiem testa datiem *NK-virknes garums* nepārsniegs $2*10^9$.

Piemēri

Ievaddati (fails <code>nk.dat</code>)	Izvaddati (fails <code>nk.rez</code>)
6 5	9
9999999 19	160

2. “SARAKSTA GARUMS”

(100 punkti)*

Sarakstu veido pēc kārtas sekojošu *elementu* virkne. Katrs elements var būt naturāls skaitlis vai saraksts. Katra saraksta pierakstu sāk ar atverošo kvadrātiekavu un beidz ar aizverošo kvadrātiekavu. Ja sarakstā ir vairāk par vienu elementu, tad blakusesošos elementus atdala ar komatiem. Citu simbolu (tajā skaitā tukšumzīmju) saraksta pierakstā nav.

Piemēram, `[1 , 2 , 55]` un `[[7] , [22 , 23] , 11]` ir sarakstu pieraksti.

Par saraksta *garumu* saucim tā elementu skaitu. Abu iepriekš doto sarakstu garums ir 3. Korekts ir arī tukšā (bezelementu) saraksta pieraksts `[]`. Tukšā saraksta garums ir 0.

Uzrakstiet programmu, kas dotam saraksta pierakstam aprēķina šī saraksta garumu!

Ievaddati

Teksta faila `saraksts.dat` pirmajā rindā dots saraksta pieraksts – simbolu virkne, kuras garums nepārsniedz 10 000 simbolu.

Izvaddati

Teksta faila `saraksts.rez` vienīgajā rindā jāizvada vesels nenegatīvs skaitlis – dotā saraksta garums.

Piemēri

Ievaddati (fails <code>saraksts.dat</code>)	Izvaddati (fails <code>saraksts.rez</code>)
<code>[[7] , [22 , 23] , 11]</code>	3
<code>[333]</code>	1
<code>[333 , [333 , 333] , [] , 5]</code>	4

*) Katra testpiemēra izpildei tiks dota viena sekunde.

3. “LAMPAS”

(100 punkti)*

Lejasbebru ciema vienīgā iela ir pilnīgi taisna un šķērso visu ciemu. Tā ir tieši M metrus gara, un dažādās vietās virs tās ir piekārtas lampas. Sākotnēji visas lampas ir ieslēgtas. Katra lampa, ja vien tā ir ieslēgta, apgaismo ielas daļu abos ielas virzienos tik tālu, cik augstu tā ir piekārtā. Ciema padome, vēloties minimizēt izmaksas, ir nolēmusi dažas lampas izslēgt, taču tā, lai nerastos neveikla situācija, ka kāds ielas posms, kas agrāk tika apgaismots, vairs netiek apgaismots (ielas posms ir apgaismots, ja katru tā punktu apgaismo vismaz viena lampa).

Uzrakstiet programmu, kas aprēķina, kādu lielāko skaitu lampu var vienlaicīgi izslēgt, ievērojot iepriekš aprakstīto noteikumu!



Ievaddati

Teksta faila `lampas.dat` pirmajā rindā doti divi naturāli skaitļi, kas atdalīti ar tukšumzīmi, – lampu skaits N ($1 \leq N \leq 100\,000$) un ielas garums metros M ($1 \leq M \leq 10^9$).

Katrā no nākamajām N faila rindām dots vienas lampas apraksts – divi veseli skaitļi x_i un h_i , kas atdalīti ar tukšumzīmi. x_i ir attālums metros no ielas sākuma ($0 \leq x_i \leq M$). h_i ir augstums metros, kādā lampa ir piekārtā ($1 \leq h_i \leq 10^9$).

Zināms, ka visas lampas ir piekārtas atšķirīgās vietās.

Izvaddati

Teksta faila `lampas.rez` vienīgajā rindā jāizvada vesels nenegatīvs skaitlis – lielākais lampu skaits, ko var vienlaicīgi izslēgt tā, ka apgaismoto ielas posmu kopgarums paliktu nemainīgs.

Piemēri

Ievaddati (fails <code>lampas.dat</code>)	Izvaddati (fails <code>lampas.rez</code>)	Piezīmes
<pre>7 11 1 1 2 2 3 2 10 1 11 2 8 1 2 4</pre>	4	Viens variants – izslēgt lampas, kas failā minētas kā 1.,2.,3. un 4. Šis variants parādīts zīmējumā. Otrs variants – izslēgt lampas, kas failā minētas kā 1.,2.,3. un 5.
<pre>5 4 0 2 1 2 2 2 3 2 4 2</pre>	4	Pietiek atstāt ieslēgtu lampu, kas failā minēta kā trešā.

*) Katra testpiemēra izpildei tiks dota viena sekunde.