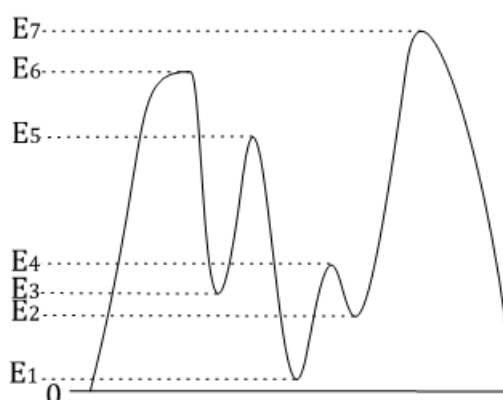


Virсотnes

Alpīnists, kas dzīvo uz kalnainas salas, ir uzkāpis kādā virсотnē, un tagad vēlas nokļūt vēl augstākā virсотnē.

Precīzāk sakot, katrs salas punkts ir pozitīvā *augstumā* virs jūras līmeņa (jūras augstums ir 0), un, ja pašreiz sasniegtās virсотnes augstums ir E_i , tad alpīnista mērķis ir sasniegt kādu virсотni augstumā E_j ($E_j > E_i$). Tā kā alpīnists atrodas virсотnē, tad nav tāda ceļa, kas uzreiz būtu vērsts augšup – lai nonāktu augstākā virсотnē, viņam vispirms nāksies nokāpt zemāk un tikai tad viņš atkal varēs kāpt augšup. Lejupiešana nekad nav tik interesanta kā ceļš augšup, tādēļ alpīnists vēlas maksimizēt zemākā punkta augstumu ceļā no pašreizējās virсотnes uz augstāku virсотni.

Piemēram, ja salas profils ir tāds, kā parādīts zīmējumā, un alpīnists atrodas virсотnē ar augstumu E_4 , tad eksistē trīs augstākas virсотnes (E_5 , E_6 un E_7), taču tieši ceļš uz E_7 ir tas, kuram zemākā punkta augstums ir vislielākais – lai šo virсотni sasniegtu, alpīnistam jānokāpj tikai līdz augstumam E_2 (visos citos gadījumos viņam nāksies nokāpt līdz augstumam E_1). Ja alpīnists uzsāktu ceļu no E_5 , atbilstošais zemākais līmenis būtu E_3 (pa ceļam uz E_6), bet, ja no E_6 , tad E_1 .



Salas karte ir divdimensionāla taisnstūrveida tabula, kas satur $N \times M$ laukumus un apraksta salas daļu augstumus – skaitlis tabulas šūnā satur atbilstošā salas reģiona augstumu. Divas šūnas atrodas kaimiņos, ja tām ir kopīgs punkts. Tādējādi, katrai šūnai (izņemot tās, kas atrodas uz tabulas robežas) ir astoņas kaimiņu šūnas. Par *ceļu* saucim šūnu virkni, kurā katras divas secīgas šūnas ir kaimiņos viena otrai. Par *līdzenu apgabalu* saucim vienas vai vairāku šūnu kopu, kuras ir vienā augstumā un kur katras divas šūnas ir savienotas ar ceļu, kas satur tikai šīs kopas šūnas. Pie tam katras divas kaimiņu virсотnes ar vienādu augstumu pieder vienam un tam pašam līdzenam apgabalam. *Virсотne* ir līdzens apgabals, kurā nevienai šūnai nav kaimiņu šūnu ar lielāku augstumu.

Uzraksties programmu, kas atrod visas salas virсотnes un katrai no tām atrod lielāko iespējamo zemākā punkta augstumu ceļā uz augstāku virсотni. Pieņemsim, ka, atrodoties kādā no salas augstākajām virсотnēm (par kurām augstāku virсотņu salā nav), alpīnists pametīs salu augstāku virсотņu meklējumos, tādēļ zemākais punkts būs 0 (jūras līmenis).

Ievaddati

Teksta faila **peaks.in** pirmā rinda satur divus veselus skaitļus N un M ($N, M \leq 2000$, $N \times M \leq 10^5$), attiecīgi, kartes augstumu un platumu. Salas kartes apraksts ir dots nākamajās N rindās. Katra no šīm rindām satur M veselus skaitļus E_{ij} ($1 \leq E_{ij} \leq 10^6$), kas atdalīti ar tukšumsimbolu. Kartes i -tās rindas un j -tās kolonas šūnas augstums E_{ij} ir dots kā j -tais skaitlis faila rindā ar numuru $i+1$.

Izvaddati

Teksta faila **peaks.out** pirmajā rindā jābūt vienam veselim skaitlim P – salas virсотņu skaitam. Katrā no nākamajām P faila rindām jābūt pa diviem veseliem skaitļiem: konkrētās virсотnes augstumam un lielākā iespējamā zemākā punkta augstumam ceļā uz augstāku

virgotni. Šī informācija jāizvada virgotņu augstuma dilstošā secībā. Ja vairākas virgotnes ir vienā augstumā, tās jāsakārto dilstošā secībā pēc attiecīgo zemāko punktu augstumiem.

1. piemērs

levaddati (fails <code>peaks.in</code>)	Izvaddati (fails <code>peaks.out</code>)	Komentārs:
<pre> 6 6 21 16 9 11 6 7 21 21 10 14 15 9 18 20 8 9 13 14 11 10 9 9 8 13 8 12 12 14 13 8 7 13 12 9 5 1 </pre>	<pre> 4 21 0 15 11 14 13 13 12 </pre>	<p>Visas virgotnes apzīmētas ar riņķiem. Ar tumšu krāsu iezīmēts viens no iespējamajiem ceļiem no virgotnes augstumā 15.</p>

2. piemērs

levaddati (fails <code>peaks.in</code>)	Izvaddati (fails <code>peaks.out</code>)
<pre> 5 3 16 14 16 14 14 15 12 17 16 12 13 10 16 11 16 </pre>	<pre> 5 17 0 16 15 16 14 16 13 16 13 </pre>

Vērtēšana

Testa piemēri, kuriem $N \leq 2$ or $M \leq 2$, ir 15 punktu vērti.

Testa piemēri, kuriem $P \leq 500$, ir 50 punktu vērti.

Testa piemēri, kuriem $P \leq 5000$, ir 80 punktu vērti.