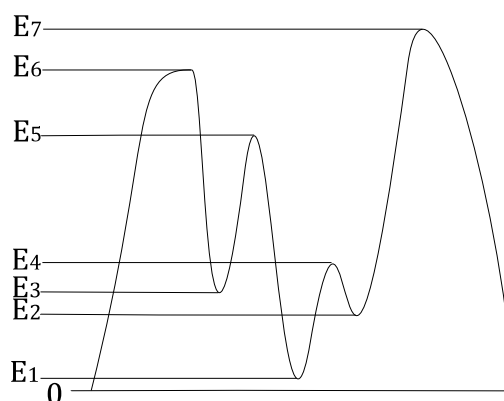


En alpinist, som bor på en bjergrig ø, er klatret til en top og ønsker nu at nå en højere top. For at være mere præcis; hvert punkt på øen har en positiv *højde* over havets overflade (havets højde er 0) og hvis toppen der er nået har højde  $E_i$ , så er alpinistens mål at nå en højere top med højde  $E_j (E_j > E_i)$ . På grund af at det er en top er der ingen direkte vej opad – for at nå et højere punkt må han først gå ned til et lavere niveau og først da kan han igen gå opad. Vejen op er ikke så bemærkelsesværdig som vejen ned – så alpinisten ønsker at maksimere højden af det laveste punkt på vejen fra den nuværende top til den højere top.

Eksempelvis hvis højdekurven af øen er som vist på billedet og alpinisten er på toppen med højde  $E_4$ , så er der tre toppe med højere højde ( $E_5$ ,  $E_6$  og  $E_7$ ), men vejen hvis laveste punkt har højeste højde er vejen til toppen med højden  $E_7$  – for at nå denne må han blot gå ned til  $E_2$  (i de andre tilfælde er han nødt til at gå ned til  $E_1$ ). Startende fra  $E_5$ , ville det laveste punkt være  $E_3$  (på vejen til  $E_6$ ), men fra  $E_6$  ville det være  $E_1$ .

Kortet over øen er en to-dimensionel rektangulær tabel, der indeholder  $N \times M$  felter, og det beskriver højde af specifikke dele af øen – nummeret i et felt beskriver højden af det tilsvarende område af øen. To felter er naboer, hvis de deler et punkt. Derfor har hvert felt (på nær felter i kanten) otte naboer. En vej er en følge af felter, hvor hver to på hinanden følgende felter er naboer. Et *fladt område* er et sæt af en eller flere felter, som alle har samme højde, og hvor ethvert par af felter fra sættet er forbundet af en vej, der kun besøger felter, der er en del af sættet. To nabofelter med samme højde tilhører samme flade område. En *top* er et fladt område, der kun har nabofelter med lavere højde.

Skriv et program der finder alle toppe på øen og for hver af dem finder højden af det højest mulige laveste punkt på vejen til en højere top. For den højeste top på øen (for hvilken der ikke er en højere top på øen) antager vi at alpinisten må forlade øen for at lede efter højere toppe og derfor vil det laveste punkt være 0 (højden af havets overflade).



### Input data

Den første linje i tekstfilen **peaks.in** indeholder to positive heltal  $N$  og  $M$  ( $N, M \leq 2000$ ,  $N \times M \leq 10^5$ ), henholdsvis højden og bredden af kortet. De følgende  $N$  linjer indeholder beskrivelsen af kortet over øen. Hver af disse linjer indeholder  $M$  heltal  $E_{ij}$  ( $1 \leq E_{ij} \leq 10^6$ ) adskilt med mellemrum. Højden af feltet  $E_{ij}$  (svarende til den  $i$ 'te række og  $j$ 'te kolonne på kortet) er givet ved det  $j$ 'te heltal i den  $i+1$ 'te linje i filen.

### Output data

Den første linje i tekstfilen **peaks.out** skal indeholde et heltal  $P$ , antallet af toppe fundet på øen. De følgende  $P$  linjer skal hver indeholde to heltal: højden af den denne top og højden af det højest mulige laveste punkt på vejen til en højere top. Informationerne omkring toppene skal skrives i faldende orden i forhold til deres højde; hvis flere toppe har samme højde skal de sorteres i faldende orden i forhold til det laveste punkts højde.

Input data (file <code>peaks.in</code> )	Output data (file <code>peaks.out</code> )	Kommentar:
<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>Alle toppe er markeret med cirkler.              En af de mulige veje fra toppen med højden 15 er vist med sort.</p>

**Example 2**

Input data (file <code>peaks.in</code> )	Output data (file <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>

**Grading**

Der gives 15 point for testsæt hvor  $N \leq 2$  eller  $M \leq 2$ .

Der gives 50 point for testsæt hvor  $P \leq 500$ .

Der gives 80 point for testsæt hvor  $P \leq 5000$ .