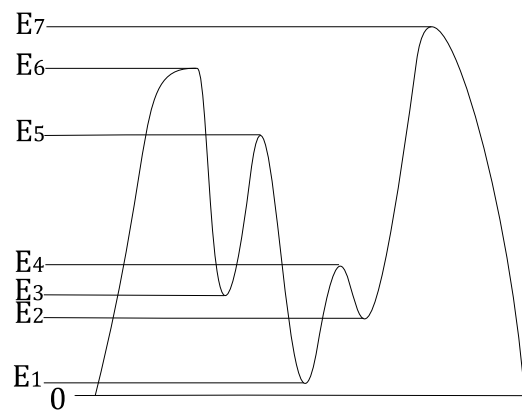


Peaks

Ein Bergsteiger, der auf einer bergigen Insel lebt, befindet sich auf einem Gipfel, und möchte nun zu einer beliebigen höher gelegenen Stelle des Eilands gelangen.

Jeder Punkt der Insel hat eine positive *Höhe* über dem Meeresspiegel (welcher auf Höhe 0 ist). Wenn der Bergsteiger also aktuell an einem Punkt mit Höhe E_i steht, dann möchte er einen Punkt erreichen mit Höhe E_j mit $E_j > E_i$. Da er bereits auf einem Gipfel steht, gibt es keinen Weg, der direkt nach oben führt. Um an einen höheren Punkt zu gelangen, muss der Bergsteiger also erst bergab, und später erst wieder berghoch gehen. Because of peak there is no immediate path uphill - to get to higher point he first needs to go downhill to some lower level and only then he can again go uphill. Da es unten niemals so schön ist wie oben möchte der Bergsteiger die Höhe des tiefsten Punktes seines Weges maximieren.

Wenn die Insel also beispielsweise den Querschnitt wie im Bild aufweist und der Bergsteiger am Gipfel der Höhe E_4 startet, dann gibt es drei höhere Gipfel (E_5 , E_6 und E_7). Und der Weg der am tiefsten Punkt am höchsten liegt ist der zur Höhe E_7 . Um dort hin zu gelangen muss er lediglich auf die Höhe E_2 hinabsteigen (in den anderen Fällen wäre er gezwungen bis auf die Höhe E_1 herunter zu gehen). Wenn er bei E_5 startet, wäre sein tiefster Punkt E_3 (auf dem Weg zu E_6), und von E_6 aus müsste er E_1 durchqueren.



Die Karte der Insel ist ein Rechteck bestehend aus $N \times M$ quadratischen Feldern, die jeweils eine Höhe besitzen. Zwei Felder sind benachbart, falls sie einen gemeinsamen Punkt haben. Also hat jedes Feld (außer am Rand) acht Nachbarn. Ein Pfad ist eine Folge von Feldern, bei der jedes Feld benachbart zu seinem Vorgänger ist. Eine *ebene Fläche* ist eine Menge von einem oder mehreren Feldern, die alle die gleiche Höhe haben und miteinander verbunden sind. Ein *Gipfel* ist eine ebene Fläche, die an keine Felder mit größerer Höhe grenzen.

Schreibe ein Programm, das alle Gipfel der Insel findet, und für jeden Gipfel einen höheren Gipfel, sodass der Weg dorthin einen möglichst hohen tiefsten Punkt hat. Für den höchsten Gipfel der Insel (für den es keinen höheren Gipfel gibt) nehmen wir an, dass der Bergsteiger die Insel verlässt auf der Suche nach höheren Bergen. Folglich ist der tiefste Punkt die Meereshöhe

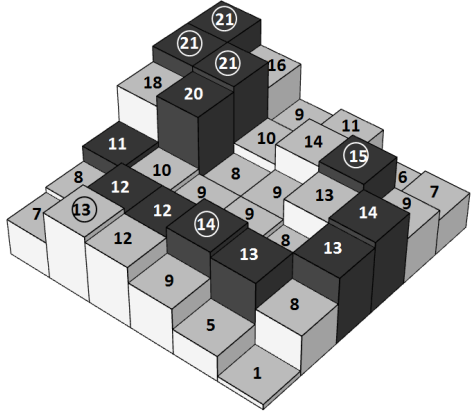
Eingabe

In der ersten Zeile der Datei **peaks.in** stehen zwei ganze Zahlen N and M ($N, M \leq 2000$, $N \times M \leq 10^5$), die Länge und Breite der Insel. In den nächsten N Zeilen steht die Höhenkarte der Insel. In jeder dieser Zeilen stehen M ganze Zahlen E_{ij} ($1 \leq E_{ij} \leq 10^6$) mit Leerzeichen getrennt. Die Höhe des Feldes E_{ij} entspricht der i -ten Zeile und der j -ten Spalte auf der Karte und findet sich als j -te Zahl in der $i+1$ -ten Zeile der Datei.

Ausgabe

In die ersten Zeile der Datei **peaks.out** muss eine ganze Zahl P, die Anzahl der gefundenen Gipfel geschrieben werden. In den folgenden P Zeilen sollen jeweils zwei Zahlen stehen: Die Höhe des Gipfels, und die (möglichst große) Höhe des tiefsten Punktes auf dem Weg zu einem höheren Gipfel. Die Gipfel müssen nach absteigender Höhe sortiert werden. Bei gleicher Höhe mit absteigender Höhe des tiefsten Punktes des zugehörigen Pfades.

Beispiel 1

| Eingabe (peaks.in) | Ausgabe (peaks.out) | Kommentar: |
|---|---------------------------------------|--|
| | |  <p>Alle Gipfel sind mit Kreisen markiert. Außerdem ist ein möglicher Pfad vom Gipfel mit Höhe 15 aus markiert.</p> |
| <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> | |

Beispiel 2

| Eingabe (peaks.in) | Ausgabe (peaks.out) |
|---|---|
| <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> |

Bewertung

Testfälle mit $N \leq 2$ oder $M \leq 2$ geben 15 Punkte.
 Testfälle mit $P \leq 500$ geben 50 Punkte.
 Testfälle mit $P \leq 5000$ geben 80 Punkte.