Time: 30 minutes

In the annual cultural program, school ABC decides to organize dances in pairs. Every pair of students (one boy, one girl) who wants to dance must register in advance. School regulations limit each boy-girl pair to at most m dances together, and limit each student to at most n dances overall. We want to maximize the number of dances. (Usually n is larger than m.)

This is a binary assignment problem for the set X of girls and the set Y of boys.

We construct a flow network G = (V, E) with vertices $X \cup Y \cup \{s, t\}$ and the following edges:
- an edge $s \to x$ with capacity n for each $x \in X$
- an edge $y \to t$ with capacity n for each $y \in Y$
- (if x and y registered to dance) an edge $x \to y$ with capacity m for each $x \in X$ and $y \in Y$
- If x and y didn't register to dance, edge $x \to y$ has a capacity 0

Because all the edges have integer capacities, the max-flow algorithm constructs an integer maximum flow f*, which can be decomposed into the sum of |f*| paths of the form $s \to x \to y \to t$ for some $x \in X$ and $y \in Y$. For each such path, we report the pair (x, y). Thus, the pair (x, y) can dance exactly $f(x \to y)$ times.

Input consists of two integers m and n in the first line.
In the second line, two integers denote |X| and |Y|.
In the third line, one integer p denotes the number of dance pairs.
In the next p lines, the dance pairs (x,y) are given where $0 \le x < |X|$ and $0 \le y < |Y|$

In output, print all the possible (x,y) pairs and the number of dances we can allow.

Sample Input:

```
3 4
7 8
7
0 3
4 5
6 2
1 7
3 5
4 6
2 4
```

Sample output:

(0,3) -> 3 dances
(4,5) -> 1 dance
(6,2) -> 3 dances
(1,7) -> 3 dances
(3,5) -> 3 dances
(4,6) -> 3 dances
(2,4) -> 3 dances