Problem D

MANHATTAN HEAT

File name: `D.{java,c,cpp}`

Brett is visiting NYC, but gets caught in the middle of the humid heat. There are N notable locations in Manhattan that have air conditioning, he'd like to visit M of them. Traveling between the locations requires him to be outside in the heat. The Cartesian coordinates of the N locations are given, which M of them, and in what order, should he visit so that he is outside for the shortest distance? Remember that in Manhattan, you can only travel north and south or east to west. Also, suppose that Brett can start at and leave from any one of the N locations (he has a taxi drop him off and pick him up).

Program Input

The input format is as follows:

```
T
N M
x0 y0
x1 y1
...
xN-1 yN-1
...
```

Where T ($1 \le T \le 20$) is the number of test cases, and x_i, y_i ($1 <= x_i, y_i <= 1000$) are the x and y coordinates of each of the N ($1 \le N \le 8$) notable locations, M ($1 \le M \le N$) of which Brett would like to visit.

Program Output

The output for each test case should be on its own line, in the following format:

```
L0 L1 L2 ... LM-1
```

Where L_o is the index (0-based) of the location that Brett should visit in order from left to right. If more than one route is minimal, use the one that comes *lexicographically first*.

P	E	A	B	O	D	Y	
P	E	A	N	U	T		
P	E	A	N	U	T	T	Y
0	1	2	3	4	5	6	7

We call the above words lexicographically ordered from top to bottom because, in ASCII, B < N from column 3 and peanutty is longer than peanut. We call peabody lexicographically first. *Many library sorts such as Java's String comparator, Arrays.sort(), Collections.sort() and C++'s sort() order strings lexicographically by default.*

Sample Input & Output

Input	Output
2 4 3 8 3 6 4 18 3 12 12 4 2 10 11 15 20 16 6 9 8	1 0 2 0 3