# Bytelandian Telecom

King Johnny has a serious drink problem, which has recently become the focus of attention of all Bytelandian tabloids and colour magazines. In a desperate effort to divert the public's attention and ingratiate himself with his subjects, he decides to start giving out valuable gifts. This time he has chosen to harass the peaceful life of the CEO of Bytelandian Telecom, and requested him to create a Metropolitan Area Network for the citizens of the capital of Byteland, as part of an "our King is a good man" campaign.

The CEO has no choice but to obey the orders he receives. This rational and business-minded man would obviously like to perform the installation at the smallest possible cost, and he asks you for your help.

The King has stated the topology of the network plainly enough in the form of a graph (not necessarily connected), with vertices corresponding to nodes (computers), and edges to the cable connections between them. It is now your task to select the points of the city to place the nodes of the network at. The city is a regular mesh of streets (depicted as vertical and horizontal segments on a map), with crossroads located at points with integer coordinates. Nodes may only be located at crossroads of streets (no two nodes at the same crossroad). Cables may only run along streets and must connect nodes by the shortest possible route under this constraint. Moreover, a cable of precisely such length must be currently in stock (you are provided with a list of possible cable lengths). Try to layout the network in such a way as to minimise the total length of cable used.

## Input

The input starts with a line containing integer $t$ <= 1000, the number of test cases. $t$ test cases follow.

The first line of a test cases begins with integer $k$, denoting the number of different available cable lengths, followed by $k$ space separated integers $p_j$ corresponding to the allowed lengths of cables (1 <= $k$ <= 100, 1 <= $p_j$ <= 100). The next line contains two integers $n$ $m$, denoting the required number of nodes and cables in the network, respectively (1 <= $n$ <= 100, 1 <= $m$ <= 1000). The next m lines contain a pair of integers $a_j$ $b_j$ each, signifying that nodes $a_j$ and $b_j$ should be connected by a cable (1 <= $a_j$,$b_j$ <= $n$).

## Output

A valid solution to the $i$-th test case consists of a line with the text 'city $i$ Y', followed by $n_i$ lines each containing two integers, the x- and y-coordinates of successive nodes in the solution (0 <= $x$,$y$ <= 100).

It is guaranteed that for every test case there exists at least one possible solution. You can however leave out a test case by outputting the line 'city $i$ N' instead of a valid solution.

## Score

For each correctly solved test case you are awarded ($m$/$sum$) * (($p_1$+$p_2$+...+$p_k$)/$k$) points, where *sum* is the total length of all cables used.

The score awarded to your program is the sum of scores for individual test cases.

## Example

**Input:**

```
4
2 1 2
4 5
1 2
2 3
3 4
1 4
2 4
1 2
4 5
1 2
2 3
3 4
1 4
2 4
2 1 2
5 8
1 2
1 3
1 4
1 5
2 4
2 5
3 4
3 5
1 1
2 1
1 2
```

**Output:**

```
city 1 Y
0 0
0 1
1 1
1 0
city 2 Y
2 0
1 1
0 2
0 0
city 3 Y
0 1
0 2
1 1
1 2
0 0
city 4 N
```

**Score:**

```
score = 3.340003
```

*Bonus info:* If score = *xxx.xxxaaa*, *aaa* means the number of test cases with non-zero score...