Problem H - Harrowing Escape

Daniel is playing a Tomb Robber game that involves looking for treasure in a pyramid.

Daniel has just found the treasure room at the end of a long corridor. The room holds more rare jewels and extravagant treasures than he can carry back! However, as Daniel leaves, he triggered a trap and caused the corridor to start collapsing! Pieces of the ceiling above have fallen onto the corridor, blocking Daniel's path.

We can model Daniel's bag as a circle. He needs to dump some of his treasures to make his bag smaller in order to fit through the fallen debris.

Help Daniel figure out how big his backpack can be to still be able to get out of the corridor alive.
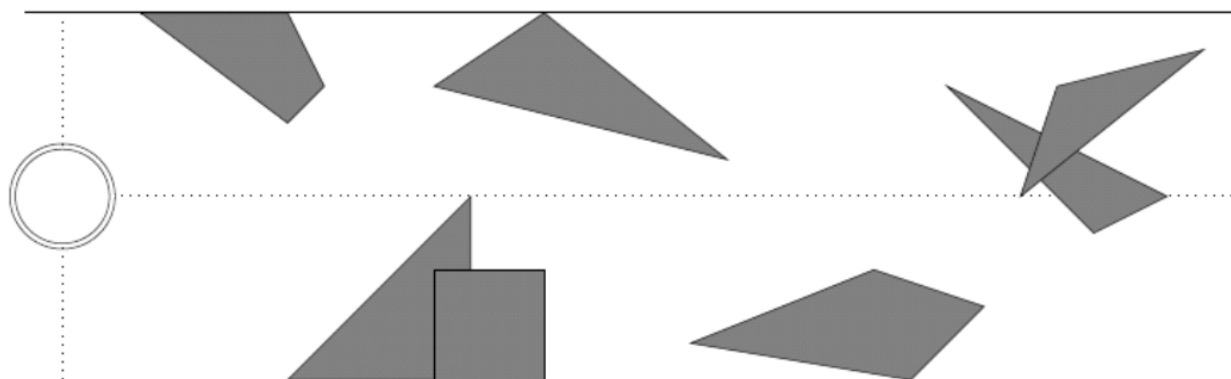

Figure 1: A diagram of sample input 1 – Your backpack is the circle on the left side and you must chart a course to the right side of the corridor.

Input

The first line contains a single integer T, the number of test cases.

Each test case begins with an even integer $2 \leq w \leq 1000$, the width of the corridor (i.e. the upper and lower boundaries are $y = \pm w/2$), and an integer $0 \leq N \leq 100$, the number of pieces of fallen debris.

The next N lines each contains the description of a single debris as a simple polygon. Each of these lines begins with a single integer $3 \leq n \leq 10$, the number of vertices, followed by n pairs of integers x, y ($0 \leq x \leq 1000$ and $-w/2 \leq y \leq w/2$) specifying the coordinates of the vertices in counterclockwise order.

Initially Daniel is at $(0, 0)$ and is guaranteed not overlap any debris. The vertices of each polygon are unique, no two non-consecutive edges of the polygon intersect or touch, and each polygon has non-zero area.

Output

For each input test case, output, rounded to two decimal places, the maximum radius $r > 0$ such that a circular backpack of that radius can move from $(0, 0)$ to the far right of the corridor without hitting any debris. Output "Impossible" if no such radius exists.

You can assume that the answer will not lie with 10^{-6} of an odd multiple of 0.005 so default rounding in your programming language will give the correct answer.

Sample Input

```
3
6 2
4 3 -1 5 -1 5 1 3 1
3 4 0 7 -1 7 1
8 2
3 2 -1 5 -1 5 4
3 4 -4 7 1 4 1
10 7
4 1 5 5 2 6 3 5 5
3 5 -5 10 -5 10 0
4 9 -5 12 -5 12 -2 9 -2
3 9 3 17 1 12 5
4 22 -5 24 -3 21 -2 16 -4
3 23 3 27 -1 29 0
3 25 0 30 4 26 3
```

Sample Output

```
1.00
Impossible
1.33
```