Riding the Fences

Farmer John owns a large number of fences that must be repaired annually. He traverses the fences by riding a horse along each and every one of them (and nowhere else) and fixing the broken parts.

Farmer John is as lazy as the next farmer and hates to ride the same fence twice. Your program must read in a description of a network of fences and tell Farmer John a path to traverse each fence length exactly once, if possible. Farmer J can, if he wishes, start and finish at any fence intersection.

Every fence connects two fence intersections, which are numbered inclusively from 1 through 500 (though some farms have far fewer than 500 intersections). Any number of fences (>=1) can meet at a fence intersection. It is always possible to ride from any fence to any other fence (i.e., all fences are "connected").

Your program must output the path of intersections that, if interpreted as a base 500 number, would have the smallest magnitude.

There will always be at least one solution for each set of input data supplied to your program for testing.

PROGRAM NAME: fence

INPUT FORMAT

Line 1:	The number of fences, F (1 <= F <= 1024)
Line 2..F+1:	A pair of integers (1 <= i,j <= 500) that tell which pair of intersections this fence connects.

SAMPLE INPUT (file fence.in)

```
9
1 2
2 3
3 4
4 2
4 5
2 5
5 6
5 7
4 6
```

OUTPUT FORMAT

The output consists of F+1 lines, each containing a single integer. Print the number of the starting intersection on the first line, the next intersection's number on the next line, and so on, until the final intersection on the last line. There might be many possible answers to any given input set, but only one is ordered correctly.

SAMPLE OUTPUT (file fence.out)

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

Submission file Name: [Choose File] No file chosen   [Send it in!]