British Informatics Olympiad Final

12–14 April, 2002

Sponsored by Lionhead Studios

Dressage for Pigeons

It is hard to take any sport quite so seriously as Pigeon Racing Up North. To enter this cuthroat world lightly, to disregard its ancient traditions or to bring in an untrained bird, is to enter fettered – a mistake few make and none repeat. From time immemorial the first event has been the dressage, to test the oneness of man and pigeon, and ensure the manoeuverability and stamina of the bird in the events to follow.

Waypoints are marked out. The pigeon has to be guided between waypoints, visiting each one in turn before returning to the start. A pigeon's skill is determined by ensuring it flies in straight lines, turning only at the specified waypoints. Tradition (and Rule 12, Section 4b) dictates that the bird cannot fly through the same airspace more than once (except at the end of their flight). Given the waypoints it is necessary for the pigeon's master to uphold tradition, by selecting an appropriate order for their visitation.

For example, consider the case (pictured above) where 1, 2 & 3 all have the same Y co-ordinate. The sequence $4 \to 1 \to 2 \to 5 \to 3 \to 4$ is invalid since $3 \to 4$ crosses airspace used by $2 \to 5$. Similarly, $1 \to 3 \to 5 \to 2 \to 4 \to 1$ is invalid because $5 \to 2$ touches the airspace used by $1 \to 3$. The sequence $1 \to 4 \to 5 \to 3 \to 2 \to 1$ is valid.

Write a program that calculates a valid path for a pigeon. The first line of the input will be a single integer n ($3 \leq n \leq 5,000$) indicating the number of waypoints to follow. Each of the next n lines will contain two integers, indicating the X then the Y co-ordinate of the waypoint. All co-ordinates will be integers between 0 and 50,000 (inclusive). The waypoints will not all lie on the same line.

Your output should consist of $n + 1$ lines indicating the order in which the waypoints should be visited. The first and last waypoints should be the same. All the waypoints must be visited.

Sample Input

```
5
10 20
20 20
30 20
15 10
25 10
```

Sample Output

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