## Programming Part

# Cycle Detection (20 pts)

## Problem Description

Given a "linked list" where each node contains $\{data, next\}$ and *next* points to another node, run the Floyd Cycle Detection algorithm below to detect whether there is a reachable cycle from the *head*. You need to print out all the data within the nodes that the hare visited before (including) meeting the tortoise.

Note that there are many different variant versions of the Floyd Cycle Detection algorithm. You are asked to implement the version below *exactly* to produce the correct answer for this problem.

FLOYD-CYCLE-DETECTION(*head*)

```
1  hare = head
2  tortoise = head
3  while hare ≠ NIL and hare.next ≠ NIL
4        hare = hare.next.next
5        tortoise = tortoise.next
6        if hare = tortoise
7              return TRUE
8  return FALSE
```

## Input

The first line includes one integer $N$, representing the number of nodes. The $n$-th line of the next $N$ lines contains two integers, which means *data* and *next* for node $n$. We assume that *head* initially points to node 1, and a *next* of 0 means connecting to NIL.

## Output

- If there is a reachable cycle from the *head*, output a line of

  ```
  [head.data] [head.next.next.data] [head.next.next.next.next.data] ... [finish.data]
  ```

  where `finish` is the location where the hare and tortoise meet.
- If there is no reachable cycle from the *head*, output a line of

  ```
  [head.data] [head.next.next.data] [head.next.next.next.next.data] ... [goodbye.data]
  ```

  where `goodbye` is the last node that the hare jumps into before leaving the linked list from the *head*.

## Constraint

- $1 \leq N \leq 10^5$
- $-2^{30} \leq data \leq 2^{30}$
- $0 \leq next \leq N$

## Sample Testcases

### Sample Input 1

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

### Sample Output 1

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

### Sample Input 2

```
3
11 2
26 2
64 1
```

### Sample Output 2

```
11 26
```

### Sample Input 3

```
5
1 2
2 3
3 4
4 0
5 3
```

### Sample Output 3

```
1 3
```

## Hint

- By design, you can pass this homework by simulating the Floyd Cycle Detection algorithm properly. There is no need for other arithmetic calculations or cuts.