import java.util.Scanner;

public class Ring

 {

    int noOfProcesses;

    Process[] processes;

    Scanner sc;

    public Ring() {

        sc = new Scanner(System.in);

    }

    public void initialiseRing() {

        System.out.println("Enter no of processes");

        noOfProcesses = sc.nextInt();

        processes = new Process[noOfProcesses];

        for (int i = 0; i < processes.length; i++) {

            processes[i] = new Process(i);

        }

    }

    public int getMax() {

        int maxId = -99;

        int maxIdIndex = 0;

        for (int i = 0; i < processes.length; i++) {

            if (processes[i].active && processes[i].id > maxId) {

                maxId = processes[i].id;

                maxIdIndex = i;

            }

        }

        return maxIdIndex;

    }

    public void performElection() {

        System.out.println("Process no " + processes[getMax()].id + " fails");

        processes[getMax()].active = false;

        System.out.println("Election Initiated by");

        int initiatorProcesss = sc.nextInt();

        int prev = initiatorProcesss;

        int next = prev + 1;

        while (true) {

            if (processes[next].active) {

                System.out.println("Process " + processes[prev].id + " pass Election(" + processes[prev].id + ") to"

                        + processes[next].id);

                prev = next;

            }

            next = (next + 1) % noOfProcesses;

            if (next == initiatorProcesss) {

                break;

            }

        }

        System.out.println("Process " + processes[getMax()].id + " becomes coordinator");

        int coordinator = processes[getMax()].id;

        prev = coordinator;

        next = (prev + 1) % noOfProcesses;

        while (true) {

            if (processes[next].active) {

                System.out.println("Process " + processes[prev].id + " pass Coordinator(" + coordinator

                        + ") message to process " + processes[next].id);

                prev = next;

            }

            next = (next + 1) % noOfProcesses;

            if (next == coordinator) {

                System.out.println("End Of Election");

                break;

            }

        }

    }

    public static void main(String arg[]) {

        Ring r = new Ring();

        r.initialiseRing();

        r.performElection();

    }

}

public class Process {

    public int id;

    public boolean active;

    public Process(int id) {

        this.id = id;

        active = true;

    }

}