```
public class Processes
{
      int processId;
    boolean active;

    public Processes(int processId) {
        this.processId = processId;
        active = true;
    }



}


import java.util.Scanner;


public class Ring
{
      Scanner input;
      Processes[] process;
      int nop;

      public Ring()
      {
            System.out.println("Ring Algorithm");
            input= new Scanner(System.in);
      }
      public void getinput()
      {
             System.out.println("Enter number of process in ring: ");
              nop = input.nextInt();
              process = new Processes[nop];

              for (int i = 0; i < nop; i++) {
                  System.out.print("Enter Process ID of p" + i + ": ");
                  int pid = input.nextInt();
                  initializeProcess(i, pid);
              }
              sortProcess();
              putOutput();
      }

      private void putOutput()
      {
             System.out.println("Processes in the ring: ");
              for(int i = 0; i < nop; i++){
                  System.out.print(process[i].processId +", active: "+
process[i].active);
                  if(i == getMax()){
                      System.out.print(", Coordinator\n");
                  }else {
```

```
                    System.out.print("\n");
                }

            }


    }
    private int getMax() {
          int max = 0, indexOfMax = 0;
      for(int i = 0; i < nop; i++){
          if(process[i].active && max <= process[i].processId ) {
              max = process[i].processId;
              indexOfMax = i;
          }
      }
      return indexOfMax;

    }
    private void sortProcess() {
          for (int i = 0; i < nop - 1; i++) {
          for (int j = 0; j < (nop - i) -1; j++) {
              if (process[j].processId > process[j + 1].processId) {
                  int temp = process[j].processId;
                  process[j].processId = process[j + 1].processId;
                  process[j + 1].processId = temp;

              }
          }
      }

    }
    private void initializeProcess(int i, int pid) {
          process[i]=new Processes(pid);

    }
      public void conductElection() {

            try{
                Thread.sleep(2000);
            }catch(Exception e ){
                System.out.println(e);
            }
            System.out.println("process "+ process[getMax()].processId
+" Fail");
            process[nop-1].active = false;

            while(true){
                System.out.print("Conduct Election?\nyes or exit: ");
                String choice = input.next();
                if("yes".equals(choice) || "Yes".equals(choice) ||
"y".equals(choice) || "Y".equals(choice)){
                    System.out.println("Election initiated by: ");
                    int initiatorProcess = input.nextInt();
                    for(int i = 0; i< nop; i++){
```

```
                        if(process[i].processId == initiatorProcess){
                            initiatorProcess = i;
                            break;
                        }
                    }
                    int prev = initiatorProcess;
                    int next = prev+1;

                    while(true){
                        if(process[next].active) {
                            System.out.println("Process "+
process[prev].processId +" pass message to process
"+process[next].processId );
                            prev = next;
                        }
                        next = (next+1) % nop;

                        if(next == initiatorProcess) {
                            break;
                        }
                    }
                    System.out.println("Process "+
process[getMax()].processId +" becomes coordinator");
                } else {
                    System.exit(0);
                }
            }
        }
    public static void main(String[] args)
    {
          System.out.println("BHAGYASHREE MEHTA [0802IT121014]");
      Ring ringElection = new Ring();
      ringElection.getinput();
      ringElection.conductElection();

    }

}
```