Ring Algorithm Program:

```
package ring;

import static java.lang.System.exit;
import java.util.Scanner;
public class Ring
{
   static int n,front=0,rear=0,ch,maxi=0,cord;
   static int[][] cq;
   public static void main(String[] args)
   {
      int a=1;
      Scanner scanner=new Scanner(System.in);
      System.out.println("Enter the no of processes : ");
      cord=n=scanner.nextInt();
      cq=new int[n+1][n+1];
      for(int i=1;i<=n;i++)
      {
       if (rear==0 && front==0)
          front=rear=1;
       else if(rear==n && front!=1)
          rear=1;
       else
            rear=rear+1;
       System.out.println("Enter the  process no : ");
       cq[rear][0]=scanner.nextInt();
       System.out.println("Enter the state of process : ");
       cq[rear][1]=scanner.nextInt();
      }
      display();
      while(a==1)
      {
          System.out.print("\n1.Crash \n2.Activate\n3.Display\n4.Quit");
          System.out.print("\nEnter the choice : ");
          ch=scanner.nextInt();
          switch(ch)
          {
               case 1 :
                      System.out.print("Enter the process no you want to crash : ");
                      //int x=scanner.nextInt();
                      crash(scanner.nextInt());
                      break;

               case 2 :
                      System.out.print("Enter the process no you want to activate : ");
                      activate(scanner.nextInt());
                      break;
               case 3 :
                      if(cq[cord][1]==1)
```

```
              {
                System.out.println("\n No need to start election.");
              }
              else
              {
                System.out.println("Enter the process number initiating election : ");
                int x2=scanner.nextInt();
                while(cq[x2][1]==0)
                {
                  System.out.println("Process "+x2 +" is crashed and cannot start
election");
                  System.out.println("Enter another Process :");
                  x2=scanner.nextInt();
                }
                dis(x2);
              }
              break;
          case 4:
              exit(0);
              break;
       }
       System.out.print("\nDo you want to continue : ");
       a=scanner.nextInt();
     }
  }
  public static void crash(int x)
  {
    for(int i=0;i<=n;i++)
    {
      if(cq[i][0]==x)
      {
        if(cq[i][1]==0)
        {
          System.out.println("Already crashed");
          break;
        }
        else
        {
          cq[i][1]=0;
          display();
          break;
        }
      }
    }
  }
  public static void activate(int x)
  {
    for(int i=0;i<=n;i++)
    {
      if(cq[i][0]==x)
```

```
        {
            if(cq[i][1]==1)
            {
                System.out.println("Already activated");
                break;
            }
            else
            {
                cq[i][1]=1;
                display();
                break;
            }
        }
    }
}

public static void display()
{
       System.out.print("\nProcess    : ");
       for(int i=1;i<=n;i++)
       {
           System.out.print("\tP"+i);
       }
       System.out.print("\nStatus     : ");
       for(int i=1;i<=n;i++)
       {
           System.out.print("\t"+cq[i][1]);
       }
       System.out.print("\nIdentifier : ");
       for(int i=1;i<=n;i++)
       {
           System.out.print("\t"+cq[i][0]);
       }
}
public static void dis(int x)
{
    rear=x-1;
    front=x;
    for(int i=front;i<=n;i++)
    {
        if(cq[i][1]!=0)
        {
            System.out.print(cq[i][0]);
            if(i!=n-1)
            {
                System.out.print("->");
            }
            maxi=Math.max(maxi, cq[i][0]);
        }
    }
    if(rear!=0)
      System.out.print("->");
    for(int i=1;i<=rear;i++)
    {
      if(cq[i][1]!=0)
      {
        System.out.print(cq[i][0]);
        if(i!=rear-1)
        {
          System.out.print("->");
        }
        maxi=Math.max(maxi, cq[i][0]);
      }
    }
    System.out.println("\nNew Co-ordinator is : "+maxi);
  }
}
```

Output:

```
$ javac Ring.java
$ java Ring

Enter the no of processes :
5
Enter the  process no :
1
Enter the state of process :
2
Enter the  process no :
2
Enter the state of process :
4
Enter the  process no :
3
Enter the state of process :
6
Enter the  process no :
4
Enter the state of process :
8
Enter the  process no :
5
Enter the state of process :
9

Process    :  P1     P2     P3     P4     P5
Status     :   2      4      6      8      9
Identifier :   1      2      3      4      5
1.Crash
2.Activate
```

```
3.Display
4.Quit
Enter the choice : 1
Enter the process no you want to crash : 2

Process    :   P1     P2     P3     P4     P5
Status     :    2      0      6      8      9
Identifier :    1      2      3      4      5
Do you want to continue (1/0): 1

1.Crash
2.Activate
3.Display
4.Quit
Enter the choice : 3
Enter the process number initiating election : 3
3->45->->1
New Co-ordinator is : 5

Do you want to continue (1/0): 1

1.Crash
2.Activate
3.Display
4.Quit
Enter the choice : 2
Enter the process no you want to activate : 2

Process    :   P1     P2     P3     P4     P5
Status     :    2      1      6      8      9
Identifier :    1      2      3      4      5
Do you want to continue (1/0): 0
```