import java.io.\*;

class BullyAlgorithm

{

int code, ch, crash;

int prc[];

public void election(int n) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nThe Coordinator has Crashed !");

int flag=1;

while(flag == 1)

{

crash = 0;

for(int i1 = 0; i1<n; i1++)

{

if(prc[i1] == 0)

crash++;

}

if(crash == n)

{

System.out.println("\n All Processes are Crashed \*\*\*!!!");

}

else

{

System.out.println("\n Enter the Initiator : ");

int init = Integer.parseInt(br.readLine());

if(init < 1 || init > n || prc[init - 1] == 0)

{

System.out.println("\n Invalid Initiator... Try again");

continue;

}

for(int i1 = init - 1; i1 < n; i1++)

{

System.out.println("Process "+(i1+1)+" called for Election.\n");

}

for(int i1 = init - 1; i1 < n; i1++)

{

if(prc[i1] == 0)

System.out.println("Process "+(i1+1)+" is Dead.");

else

System.out.println("Process "+(i1+1)+" is In.");

}

for(int i1 = n-1; i1 >= 0; i1--)

{

if(prc[i1] == 1)

{

code = (i1+1);

System.out.println("\n \*\*\*New Coordinator is "+code+" \*\*\*");

flag = 0;

break;

}

}

}

} // end of while

} // end of election() method

public void Bully() throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the Number of Processes : ");

int n = Integer.parseInt(br.readLine());

prc = new int[n];

crash = 0;

for(int i = 0; i < n; i++)

prc[i] =1;

code = n;

do

{

System.out.println("\n\t1. Crash A Process");

System.out.println("\t2. Recover A Process");

System.out.println("\t3. Display New Coordinator");

System.out.println("\t4. Exit");

ch=Integer.parseInt(br.readLine());

switch(ch)

{

case 1: System.out.println("\nEnter A Process To Crash");

int cp=Integer.parseInt(br.readLine());

if((cp > n) || (cp < 1))

{

System.out.println("Invalid Process! Enter A Valid Process");

}

else if((prc[cp - 1] == 1) && (code != cp))

{

prc[cp - 1] = 0;

System.out.println("\nProcess "+cp+ " Has Been Crashed");

}

else if((prc[cp - 1] == 1) && (code == cp))

{

prc[cp - 1] = 0;

election(n);

}

else

System.out.println("\nProcess "+cp+" Is Already Crashed");

break;

case 2: System.out.println("\nCrashed Processes Are: \n");

for(int i = 0; i < n; i++)

{

if(prc[i] == 0)

System.out.println(i+1);

crash++;

}

System.out.println("Enter The Process You Want To Recover");

int rp=Integer.parseInt(br.readLine());

if((rp < 1) || (rp > n))

System.out.println("\nInvalid Process. Enter A Valid ID");

else if((prc[rp - 1] == 0) && (rp > code))

{

prc[rp - 1] = 1;

System.out.println("\nProcess "+rp+" Has Recovered");

code = rp;

System.out.println("\nProcess "+rp+ " Is The New Coordinator");

}

else if(crash == n)

{

prc[rp - 1] = 1;

code = rp;

System.out.println("\nProcess "+rp+ " Is The New Coordinator");

crash--;

}

else if((prc[rp - 1] == 0) && (rp < code))

{

prc[rp - 1] = 1;

System.out.println("\nProcess "+rp+" Has Recovered");

}

else

System.out.println("\nProcess "+rp+" Is Not A Crashed Process");

break;

case 3:

System.out.println("\nCurrent Coordinator Is " +code);

break;

case 4:

System.exit(0);

break;

default:

System.out.println("\nInvalid Entry!");

break;

}//end switch

}while(ch!=4);

}//end of Bully()

public static void main(String args[]) throws IOException

{

BullyAlgorithm ob=new BullyAlgorithm();

ob.Bully();

}

}

