***BullyAlgo.java***

import java.io.\*;

class BullyAlgo {

int cood, ch, crash;

int prc[];

public void election(int n) throws IOException {

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

System.out.println("The coordinator has crashed!");

int flag = 1;

while (flag == 1) {

crash = 0;

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

if (prc[il] == 0)

crash++;

if (crash == n) {

System.out.println("All Processes Are Crashed");

break;

} else {

System.out.println("Enter the Initiator");

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

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

System.out.println("Invalid Initiator");

continue;

}

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

System.out.println("Process " + (il + 1) + " called for Election");

System.out.println();

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

if (prc[il] == 0) {

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

} else

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

}

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

if (prc[il] == 1) {

cood = (il + 1);

System.out.println("New Coordinator is " + cood + "");

flag = 0;

break;

}

}

}

}

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;

cood = n;

do {

System.out.println("1.Crash A Process");

System.out.println("2.Recover A Process");

System.out.println("3.Display New Coordinator");

System.out.println("4.Exit");

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

switch (ch) {

case 1: System.out.println("Enter 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) && (cood != cp)) {

prc[cp - 1] = 0;

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

} else if ((prc[cp - 1] == 1) && (cood == cp)) {

prc[cp - 1] = 0;

election(n);

} else

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

break;

case 2: System.out.println("Crashed Processes are : ");

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("Invalid Process. Enter a valid ID");

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

prc[rp - 1] = 1;

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

cood = rp;

System.out.println("Process " + rp + "Is the new Coordinator");

} else if (crash == n) {

prc[rp - 1] = 1;

cood = rp;

System.out.println("Process " + rp + "Is the new Coordinator");

crash--;

} else if ((prc[rp - 1] == 0) && (rp < cood)) {

prc[rp - 1] = 1;

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

} else

System.out.println("Process " + rp + " Is not a crashed process");

break;

case 3: System.out.println("Current Coordinator is " + cood);

break;

case 4: System.exit(0);

break;

default: System.out.println("Invalid Entry");

break;

}

} while (ch != 4);

}

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

BullyAlgo ob = new BullyAlgo();

ob.Bully();

}

}

**Output:**



