Name : Abhay Sharma
Roll No. 28 / Div: C
Student Id : 20102065
Subject : DC

Experiment No. 5

Code:

```
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("\nThe 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("\n*** All Processes Are Crashed ***");
            break;
         } else {
            System.out.println("\nEnter The Initiator");
            int init = Integer.parseInt(br.readLine());

            if ((init < 1) || (init > n) || (prc[init - 1] == 0)) {
               System.out.println("\nInvalid 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("\n*** New Coordinator Is " + (cood) + "
***");
                  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;
         cood = 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) && (cood != cp)) {
                     prc[cp - 1] = 0;
                     System.out.println("\nProcess " + cp + " Has Been Crashed");
                  } else if ((prc[cp - 1] == 1) && (cood == 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 > cood)) {
                prc[rp - 1] = 1;
                System.out.println("\nProcess " + rp + " Has Recovered");
                cood = rp;
                System.out.println("\nProcess " + rp + " Is The New
Coordinator");
            } else if (crash == n) {
                prc[rp - 1] = 1;
                cood = rp;
                System.out.println("\nProcess " + rp + " Is The New
Coordinator");
                crash--;
            } else if ((prc[rp - 1] == 0) && (rp < cood)) {
                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 " + cood);
            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 {
    BullyAlgo ob = new BullyAlgo();
    ob.Bully();
  }
}
```

Output:

```
apsit@apsit-HP-280-G2-SFF:~/Documents$ java BullyAlgo
Enter The Number of Processes:
6

        1. Crash A Process
        2. Recover A Process
        3. Display New Coordinator
        4. Exit
1

Enter A Process To Crash
6

The Coordinator Has Crashed!

Enter The Initiator
4
Process 4 Called For Election
Process 5 Called For Election
Process 6 Called For Election

Process 4 Is In
Process 5 Is In
Process 6 Is Dead

*** New Coordinator Is 5 ***

        1. Crash A Process
        2. Recover A Process
        3. Display New Coordinator
        4. Exit
```