Code

1. Bully.java

Bully.java > Bully > up(int)

```java

import java.util.Scanner;
public class Bully {

    static boolean state[]  = new boolean[5];
    int coordinator;
    public static void up(int up)
    {
        if(state[up-1]==true)
        {
            System.out.println("process"+up+"is already up");
        }
        else
        {
            state[up-1] = true;
            System.out.println("process "+up+" held election");
            for(int i=up;i<5;i++)
            {
                System.out.println("election message sent from process "+up+" to process"+(i+1));
            }
            for(int i=up+1;i<=5;i++)
            {
                if(state[i-1]==true)
                {
                    System.out.println("alive message send from process "+i+" to process"+up);
                    break;
                }
            }
        }
    }
    public static void down(int down)
    {
        if(state[down-1]==false)
        {
            System.out.println("process "+down+" is already dowm.");
        }
        else
        {
            state[down-1] = false;

        }
    }

    public static void mess(int mess)
    {
        if(state[mess-1]==true)
        {
            if(state[4]==true)
```

Ring.java | Bully.java

Bully.java > Bully > up(int)

```java
            if(state[4]==true)
            {
                System.out.println("OK");
            }
            else
            {
                if(state[4]==false)
                {
                    System.out.println("process"+mess+"election");
                    for(int i=mess;i<5;i++)
                    {
                        System.out.println("election send from process"+mess+"to process "+(i+1));
                    }

                    for(int i=5;i>=mess;i--)
                    {
                        if(state[i-1]==true)
                        {
                            System.out.println("Coordinator message send from process"+i+"to all")
                            break;
                        }
                    }
                }
            }
        }
        else
        {
            System.out.println("Prccess"+mess+"is down");
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;
        for(int i=0;i<5;i++)
        {
            state[i] = true;
        }
        System.out.println("5 active process are:");
        System.out.println("Process up  = p1 p2 p3 p4 p5");
        System.out.println("Process 5 is coordinator");

        do
        {
            System.out.println(".........");
            System.out.println("1 up a process.");
            System.out.println("2.down a process");
            System.out.println("3.send a message");
```

Ring.java Bully.java

Bully.java > Bully > up(int)

```java
            System.out.println("........");
            System.out.println("1 up a process.");
            System.out.println("2.down a process");
            System.out.println("3 send a message");
            System.out.println("4.Exit");
            choice = sc.nextInt();
            switch(choice)
            {
                case 1:
                {
                    System.out.println("bring proces up");
                    int up = sc.nextInt();
                    if(up==5)
                    {
                        System.out.println("process 5 is co-ordinator");
                        state[4] = true;

                    }
                    else
                    {
                        up(up);
                    }
                }
                break;
                case 2:
                {
                    System.out.println("bring down any process.");
                    int down = sc.nextInt();
                    down(down);
                }
                break;
                case 3:
                {
                    System.out.println("which process will send message");
                    int mess = sc.nextInt();
                    mess(mess);
                }
                break;

            }

        }
        while(choice!=4);
        sc.close();
    }
}
```

2. Ring.java

Ring.java × Bully.java 1

Ring.java > Ring > election(int)

```java
import java.util.Scanner;

public class Ring {

    int n, inactive_count;
    int coordinator;
    boolean[] process_state;

    public Ring(int n) {
        this.n = n;
        this.inactive_count = 0;
        this.process_state = new boolean[n];
        //  State all processes as active
        for(int i = 0; i < n; i++) {
            this.process_state[i] = true;
        }
        this.coordinator = n - 1;
        System.out.println("Process " + n + " is set as initial coordinator");
    }

    public void deactivate_process(int id) {
        /*
         *  Input    :   Process ID
         *  Utility  :   Deactivate process
         *  Output   :   None
         */
         if(id > n || id < 0) {
            System.out.println("Invalid ID");
            return;
         }
         if(!process_state[id - 1]) {
            System.out.println("Process already inactive");
         } else {
            process_state[id - 1] = false;
            System.out.println("Process " + id + " deactivated");
            inactive_count += 1;
         }
    }

    public void view_ring() {
        /*
         *  Input    :   None
         *  Utility  :   Display ring
         *  Output   :   Console output
         */

         if(this.inactive_count == n) {
            System.out.println("All members inactive...");
```

```java
            return;
        }
        System.out.println("Active Ring members");
        for(in boolean[] process_state
            if(process_state[i]) System.out.println((i + 1) + " ");
        }
    }

    public void election(int id) {
        /*
        *  Input   :   Initiator
        *  Utility :   Hold election process to select coordinator
        *  Output  :   Coordinator id
        */
        if(this.inactive_count == this.n) {
            System.out.println("All members inactive...");
            System.out.println("Aborting election process...");
            this.coordinator = -1;
            return;
        }
        id = id - 1;
        int current_coordinator = id;
        int token = (id + 1) % n;
        System.out.println("\nElection initiator : " + (id + 1));
        // Election algorithm
        while(token != id) {
            System.out.println("Token at process " + (token + 1));
            if(this.process_state[token]) {
                if(token > current_coordinator) {
                    current_coordinator = token;
                }
            }
            token = (token + 1) % this.n;
        }
        System.out.println("Elected coordinator : " + (current_coordinator + 1));
        this.coordinator = current_coordinator;
    }

    public void ping_coordinator(int id) {
        if(!this.process_state[id - 1]) {
            System.out.println("Process inactive...");
            System.out.println("Aborting...");
            return;
        }
        if(id == coordinator) {
            if(this.process_state[id - 1]) {
                System.out.println("Coordinator active");
```

Ring.java × Bully.java 1

Ring.java > Ring > election(int)

```java
            } else {
                System.out.println("Coordinator inactive!\nInitiate election from other process");
            }
        }
        System.out.println("Sending message from process " + id + " to " + (this.coordinator + 1));
        if(!this.process_state[this.coordinator]) {
            System.out.println("Coordinator process not responding");
            System.out.println("Conducting election...");
            this.election(id);
        } else {
            System.out.println("Coordinator alive");
        }
    }

    public void setCoordinator(int c) {
        this.coordinator = c;
    }

    Run | Debug
    public static void main(String[] args) {
        int choice = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of processes: ");
        int n = sc.nextInt();
        Ring ring = new Ring(n);

        while(choice < 5) {
            System.out.println("***********Menu***********");
            System.out.println("1. Deactivate a process");
            System.out.println("2. Ping coordinator");
            System.out.println("3. View Ring");
            System.out.println("4. Election");
            System.out.println("5. Exit");
            System.out.println("**************************");
            System.out.println("Enter Choice : ");
            choice = sc.nextInt();
            switch(choice) {
                case 1 : {
                    int id;
                    System.out.println("Enter process ID : ");
                    id = sc.nextInt();
                    ring.deactivate_process(id);
                    System.out.println("");
                    break;
                }
                case 2 : {
                    int id;
                    System.out.println("Enter process ID for sender");
```

```
                    System.out.println("Enter process ID for sender");
                    id = sc.nextInt();
                    ring.ping_coordinator(id);
                    System.out.println("");
                    break;
                }
                case 3 : {
                    ring.view_ring();
                    System.out.println("");
                    break;
                }
                case 4 : {
                    int id;
                    System.out.println("Enter process ID for election initiator");
                    id = sc.nextInt();
                    ring.election(id);
                    System.out.println("");
                    break;
                }
                case 5 :
                default : {
                    System.out.println("");
                    break;
                }
            }

        }
        System.out.println("Program terminated..");
        sc.close();
    }
}
```

Output

1. Bully Algorithm

```
pict@pict-OptiPlex-5060: ~/Desktop
pict@pict-OptiPlex-5060:~/Desktop$ java Bully
5 active process are:
Process up  = p1 p2 p3 p4 p5
Process 5 is coordinator
.........
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
2
process2is already up
.........
1 up a process.
2.down a process
3 send a message
4.Exit
2
bring down any process.
5
.........
1 up a process.
2.down a process
3 send a message
4.Exit
3
which process will send message
3
process3election
election send from process3to process 4
election send from process3to process 5
Coordinator message send from process4to all
.........
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
4
process4is already up
.........
1 up a process.
2.down a process
3 send a message
4.Exit
4
pict@pict-OptiPlex-5060:~/Desktop$
```

2. Ring Algorithm

```
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment6$ javac Ring.java
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment6$ java Ring
Enter number of processes:
5
Process 5 is set as initial coordinator
***********Menu***********
1. Deactivate a process
2. Ping coordinator
3. View Ring
4. Election
5. Exit
*************************
Enter Choice :
1
Enter process ID :
5
Process 5 deactivated

***********Menu***********
1. Deactivate a process
2. Ping coordinator
3. View Ring
4. Election
5. Exit
*************************
Enter Choice :
2
Enter process ID for sender
3
Sending message from process 3 to 5
Coordinator process not responding
Conducting election...
```

```
Sending message from process 3 to 5
Coordinator process not responding
Conducting election...

Election initiator : 3
Token at process 4
Token at process 5
Token at process 1
Token at process 2
Elected coordinator : 4

***********Menu***********
1. Deactivate a process
2. Ping coordinator
3. View Ring
4. Election
5. Exit
**************************
Enter Choice :
3
Active Ring members
1
2
3
4

***********Menu***********
1. Deactivate a process
2. Ping coordinator
3. View Ring
4. Election
5. Exit
**************************
Enter Choice :
5

Program terminated..
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment6$
```