Bully Code

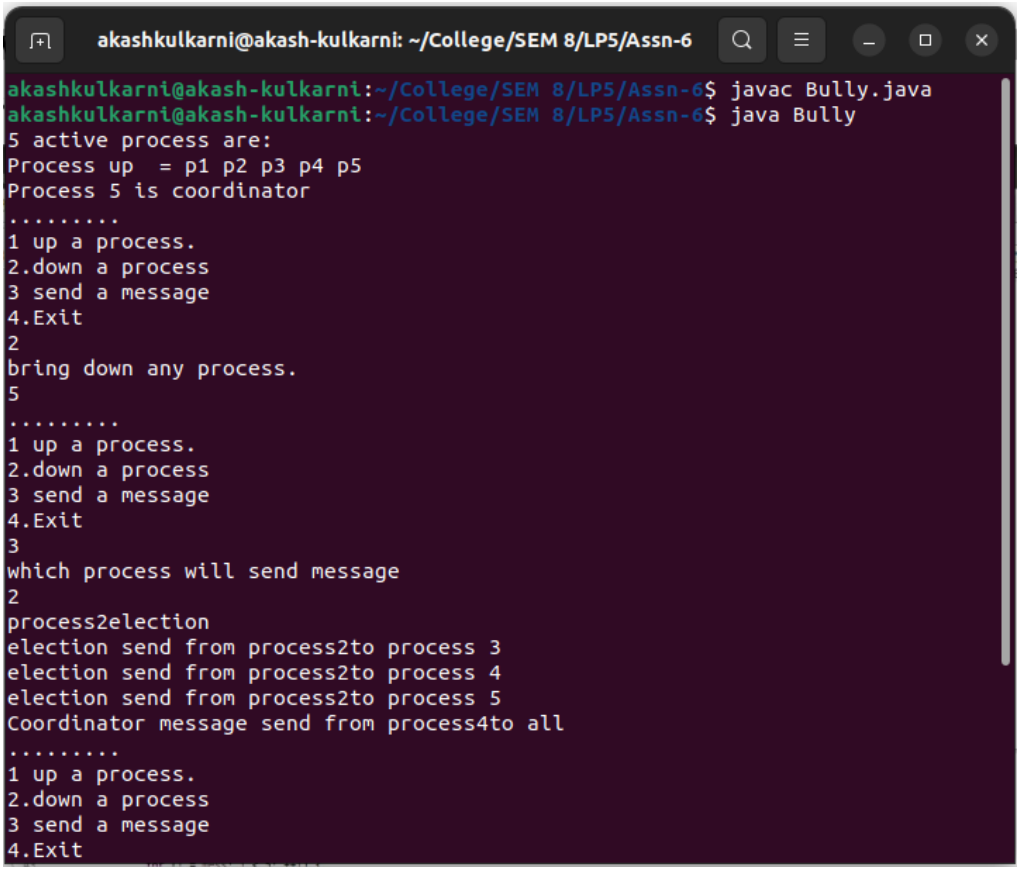
Bully.java

```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;
                }
            }
        }
    }
```

Ring.java

```java
    public static void mess(int mess)
    {
        if(state[mess-1]==true)
        {
            if(state[4]==true)
            {
                System.out.println("0K");
            }
            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");
```

Bully Output



Ring Code

```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;
```

```java
    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;
    }
```

Ring Output

```
akashkulkarni@akash-kulkarni: ~/College/SEM 8/LP5/Assn-6
akashkulkarni@akash-kulkarni:~/College/SEM 8/LP5/Assn-6$ java Ring
Enter number of processes:
3
Process 3 is set as initial coordinator
***********Menu**********
1. Deactivate a process
2. Ping coordinator
3. View Ring
4. Election
5. Exit
*************************
Enter Choice :
2
Enter process ID for sender
2
Coordinator active
Sending message from process 2 to 3
Coordinator alive

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

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

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

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