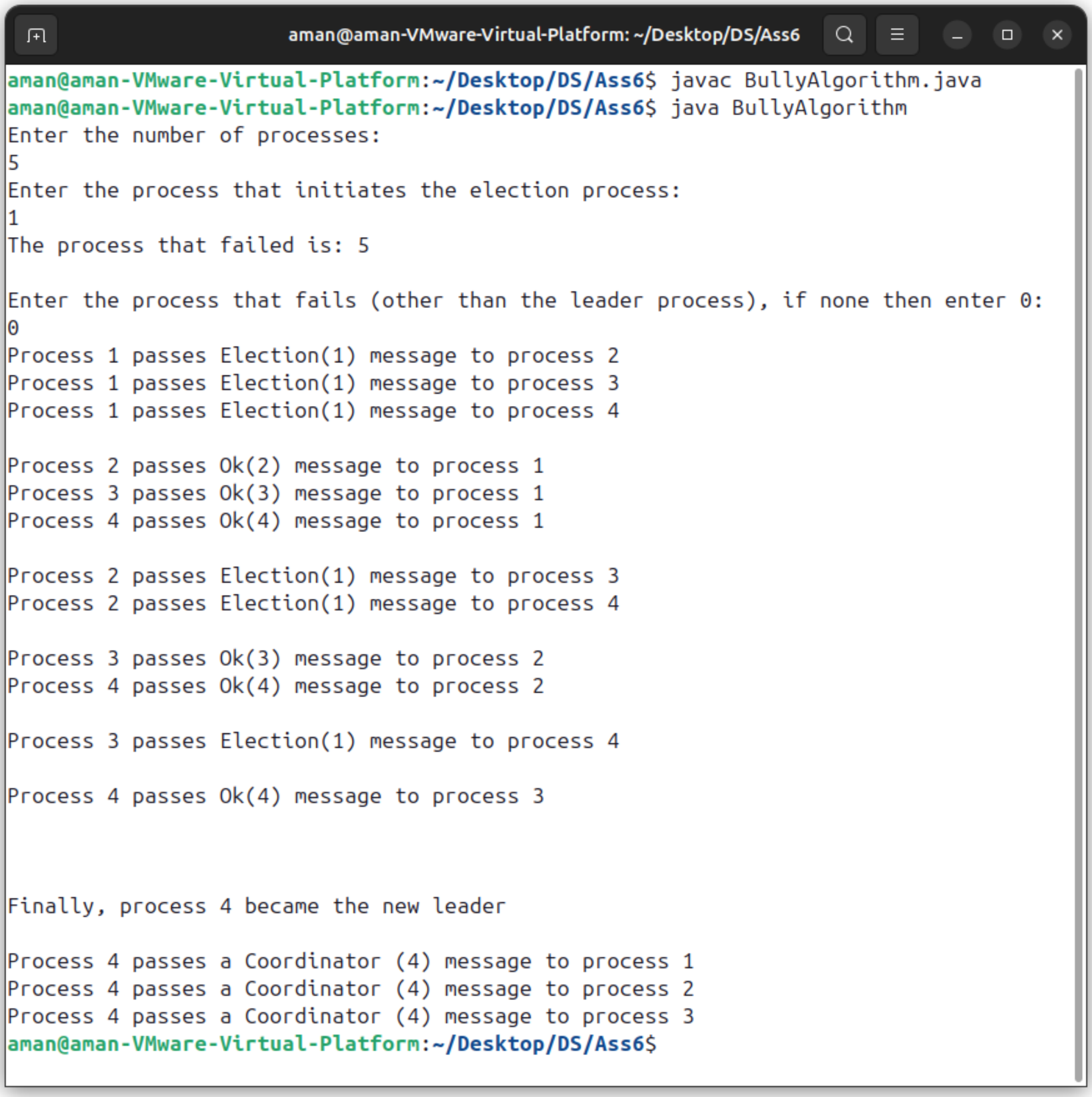
Assignment 6

1) Bully Algorithm

BullyAlgorithm.java

```
import java.util.*;

public class BullyAlgorithm {
    static int num_pr; // number of processes
    static int old_cord; // the failed coordinator or leader
    static int new_cord; // the new elected leader
    static int curr_elec; // the current process that is holding the election
    static int isActive[];
    static int failed_process;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of processes: ");
        num_pr = sc.nextInt();
        isActive = new int[num_pr + 1];

        for (int i = 1; i <= num_pr; i++) {
            isActive[i] = 1;
        }

        old_cord = num_pr;

        // Leader has failed
        isActive[old_cord] = 0;
        System.out.println("Enter the process that initiates the election process: ");
        curr_elec = sc.nextInt();
        System.out.println("The process that failed is: " + old_cord + "\n");

        System.out.println("Enter the process that fails (other than the leader process), if none then enter 0: ");
        failed_process = sc.nextInt();
        isActive[failed_process] = 0;

        // Output
        new_cord = election_process(isActive, old_cord, curr_elec);
        System.out.println("Finally, process " + new_cord + " became the new leader\n");

        // Inform all processes about the new leader
        for (int i = 1; i < num_pr - 1; i++) {
            if (isActive[i] == 1) {
                System.out.println("Process " + new_cord + " passes a Coordinator (" + new_cord + ") message to process " + i);
            }
        }

        sc.close();
    }

    public static int election_process(int isActive[], int old_cord, int curr_elec) {
        int higher_process = curr_elec;

        for (int i = curr_elec; i <= num_pr; i++) {
            if (isActive[i] == 1) {
                for (int j = i + 1; j <= num_pr; j++) {
                    if (isActive[j] == 1) {
                        System.out.println("Process " + i + " passes Election(" + curr_elec + ") message to process " + j);
                    }
                }
                System.out.println();

                for (int j = i + 1; j < num_pr; j++) {
                    if (isActive[j] == 1) {
                        System.out.println("Process " + j + " passes Ok(" + j + ") message to process " + i);
                    }
                    if (higher_process < j) {
                        higher_process = j;
                    }
                }
                System.out.println();
            }
        }
        return higher_process;
    }
}
```

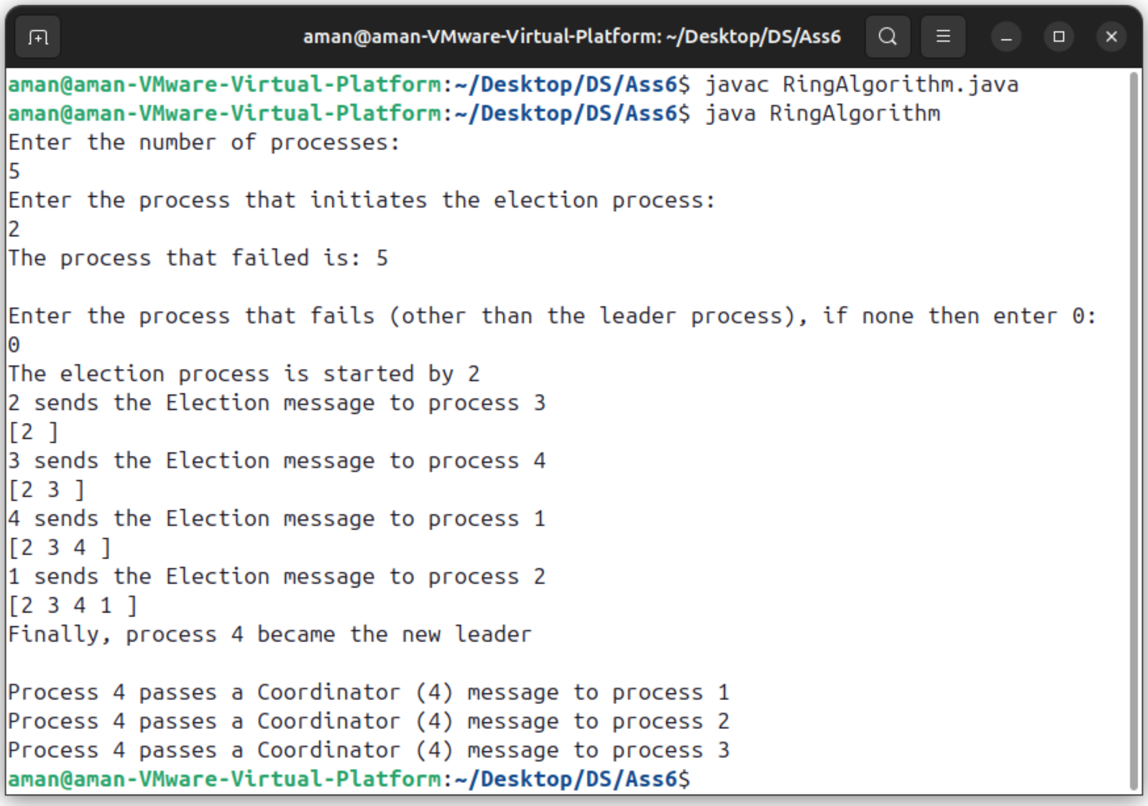


```
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$ javac BullyAlgorithm.java
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$ java BullyAlgorithm
Enter the number of processes:
5
Enter the process that initiates the election process:
1
The process that failed is: 5

Enter the process that fails (other than the leader process), if none then enter 0:
0
Process 1 passes Election(1) message to process 2
Process 1 passes Election(1) message to process 3
Process 1 passes Election(1) message to process 4

Process 2 passes Ok(2) message to process 1
Process 3 passes Ok(3) message to process 1
Process 4 passes Ok(4) message to process 1

Process 2 passes Election(1) message to process 3
Process 2 passes Election(1) message to process 4

Process 3 passes Ok(3) message to process 2
Process 4 passes Ok(4) message to process 2

Process 3 passes Election(1) message to process 4

Process 4 passes Ok(4) message to process 3



Finally, process 4 became the new leader

Process 4 passes a Coordinator (4) message to process 1
Process 4 passes a Coordinator (4) message to process 2
Process 4 passes a Coordinator (4) message to process 3
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$
```

2) Ring Algorithm

RingAlgorithm.java

```
import java.util.*;

public class RingLeaderElection {
    static int num_pr; // number of processes
    static int old_cord; // the failed coordinator or leader
    static int new_cord; // the new elected leader
    static int initiator; // the current process that is holding the election
    static int isActive[];
    static int failed_process;
    static int arr[];

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of processes: ");
        num_pr = sc.nextInt();
        isActive = new int[num_pr + 1];

        for (int i = 1; i <= num_pr; i++) {
            isActive[i] = 1;
        }

        old_cord = num_pr;

        // Leader has failed
        isActive[old_cord] = 0;
        System.out.println("Enter the process that initiates the election process: ");
        initiator = sc.nextInt();
        System.out.println("The process that failed is: " + old_cord + "\n");

        System.out.println("Enter the process that fails (other than the leader process), if none then enter 0: ");
        failed_process = sc.nextInt();
        isActive[failed_process] = 0;

        // Output
        new_cord = election_process(isActive, old_cord, initiator);
        System.out.println("Finally, process " + new_cord + " became the new leader\n");

        for (int i = 1; i < num_pr - 1; i++) {
            if (isActive[i] == 1) {
                System.out.println("Process " + new_cord + " passes a Coordinator (" + new_cord + ") message to process " + i);
            }
        }

        sc.close();
    }

    public static int election_process(int isActive[], int old_cord, int initiator) {
        System.out.println("The election process is started by " + initiator);
        int index = 0;
        arr = new int[num_pr + 1];
        int i = initiator;
        int receiver = (i % num_pr) + 1;
        while (index <= num_pr - 1) {
            if (isActive[i] == 1 && i != receiver) {
                if (isActive[receiver] == 0) {
                    receiver = (receiver % num_pr) + 1;
                }
                System.out.println(i + " sends the Election message to process " + receiver);
                arr[index] = i;
                print_array(arr, index + 1);
            }
            i = (i % num_pr) + 1;
            receiver = (i % num_pr) + 1;
            index++;
        }

        new_cord = 0;
        for (int j = 0; j <= num_pr; j++) {
            if (new_cord < arr[j]) {
                new_cord = arr[j];
            }
        }
        return new_cord;
    }

    public static void print_array(int arr[], int size) {
        System.out.print("[");
        for (int i = 0; i < size; i++) {
            if (arr[i] == 0)
                continue;
            System.out.print(arr[i] + " ");
        }
        System.out.print("]");
        System.out.println();
    }
}
```



```
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$ javac RingAlgorithm.java
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$ java RingAlgorithm
Enter the number of processes:
5
Enter the process that initiates the election process:
2
The process that failed is: 5

Enter the process that fails (other than the leader process), if none then enter 0:
0
The election process is started by 2
2 sends the Election message to process 3
[2 ]
3 sends the Election message to process 4
[2 3 ]
4 sends the Election message to process 1
[2 3 4 ]
1 sends the Election message to process 2
[2 3 4 1 ]
Finally, process 4 became the new leader

Process 4 passes a Coordinator (4) message to process 1
Process 4 passes a Coordinator (4) message to process 2
Process 4 passes a Coordinator (4) message to process 3
aman@aman-VMware-Virtual-Platform:~/Desktop/DS/Ass6$
```