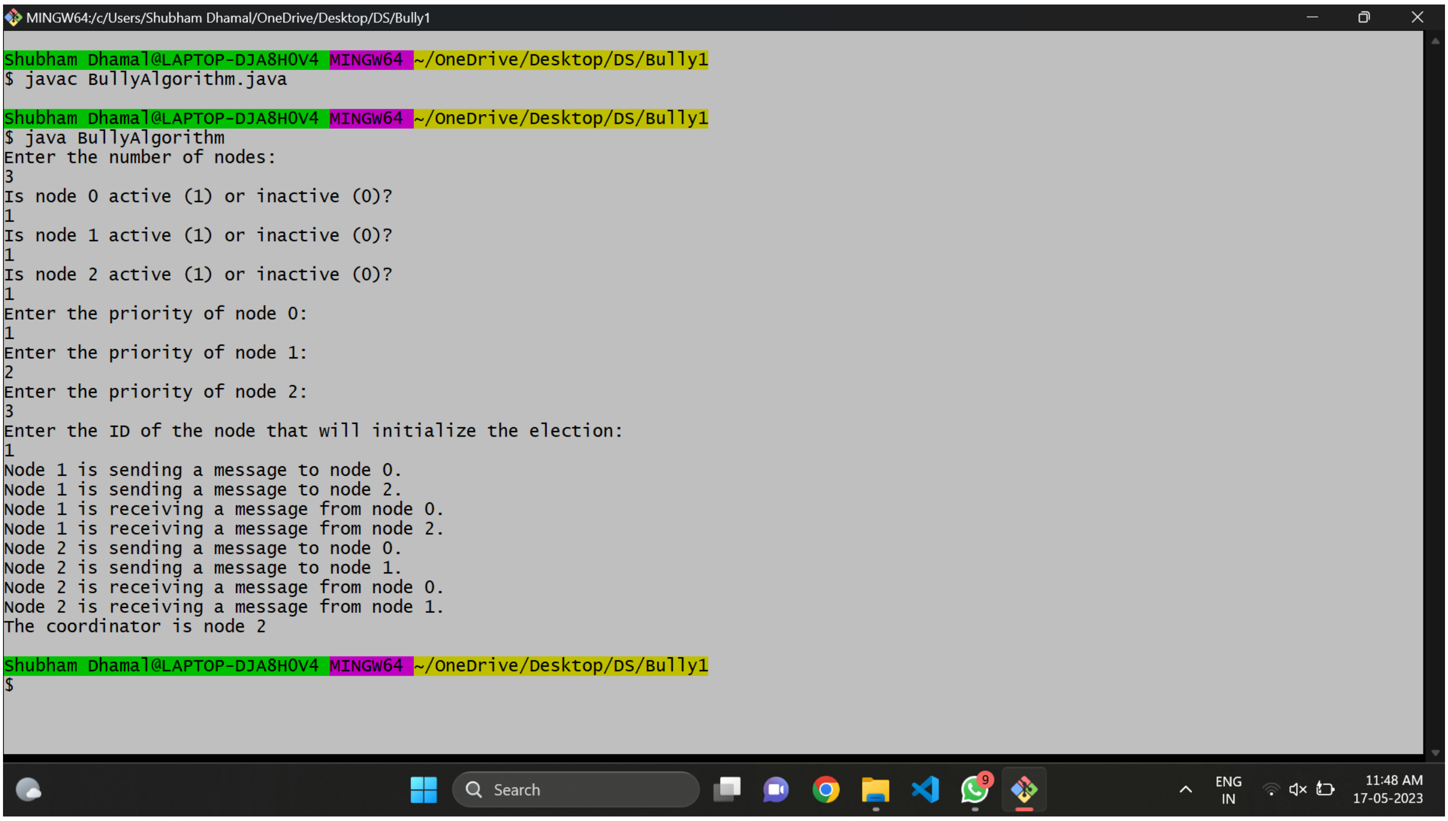
BullyAlgorithm.java

```
import java.util.Scanner;

public class BullyAlgorithm {

  private static int numberOfNodes;
  private static boolean[] processStatus;
  private static int[] processPriority;

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

    System.out.println("Enter the number of nodes:");
    numberOfNodes = sc.nextInt();

    processStatus = new boolean[numberOfNodes];
    for (int i = 0; i < numberOfNodes; i++) {
      System.out.println("Is node " + i + " active (1) or inactive (0)?");
      int nodeStatus = sc.nextInt();
      processStatus[i] = (nodeStatus == 1);
    }

    processPriority = new int[numberOfNodes];
    for (int i = 0; i < numberOfNodes; i++) {
      System.out.println("Enter the priority of node " + i + ":");
      int nodePriority = sc.nextInt();
      processPriority[i] = nodePriority;
    }

    // Initialize the leader as -1
```

```
        int leader = 1;

        // Enter the ID of the node that will initialize the election
        System.out.println("Enter the ID of the node that will initialize the election:");
        int electionInitiator = sc.nextInt();

        // Start the bully algorithm
        for (int i = electionInitiator; i < numberOfNodes; i++) {
          if (processStatus[i]) {
            // If the current process is active and it has a higher process ID and priority than the current leader,
            // then it becomes the new leader.
            if (i > leader && processPriority[i] > processPriority[leader]) {
              leader = i;
            }

            // Send a message to all other processes, informing them that it is the new leader.
            for (int j = 0; j < numberOfNodes; j++) {
              if (i != j && processStatus[j]) {
                System.out.println("Node " + i + " is sending a message to node " + j + ".");
              }
            }

            // Receive responses from all other processes.
            for (int j = 0; j < numberOfNodes; j++) {
              if (i != j && processStatus[j]) {
                System.out.println("Node " + i + " is receiving a message from node " + j + ".");
              }
            }
          }
        }
        // Print the output

        System.out.println("The coordinator is node " + leader);

    }

}
```

OUTPUT:



```
Shubham Dhamal@LAPTOP-DJA8H0V4 MINGW64 ~/OneDrive/Desktop/DS/Bully1
$ javac BullyAlgorithm.java

Shubham Dhamal@LAPTOP-DJA8H0V4 MINGW64 ~/OneDrive/Desktop/DS/Bully1
$ java BullyAlgorithm
Enter the number of nodes:
3
Is node 0 active (1) or inactive (0)?
1
Is node 1 active (1) or inactive (0)?
1
Is node 2 active (1) or inactive (0)?
1
Enter the priority of node 0:
1
Enter the priority of node 1:
2
Enter the priority of node 2:
3
Enter the ID of the node that will initialize the election:
1
Node 1 is sending a message to node 0.
Node 1 is sending a message to node 2.
Node 1 is receiving a message from node 0.
Node 1 is receiving a message from node 2.
Node 2 is sending a message to node 0.
Node 2 is sending a message to node 1.
Node 2 is receiving a message from node 0.
Node 2 is receiving a message from node 1.
The coordinator is node 2

Shubham Dhamal@LAPTOP-DJA8H0V4 MINGW64 ~/OneDrive/Desktop/DS/Bully1
$
```