Name: Abhay Sharma
Rollno:162 / Div:C
Subject:DC
Moodle id:20102065

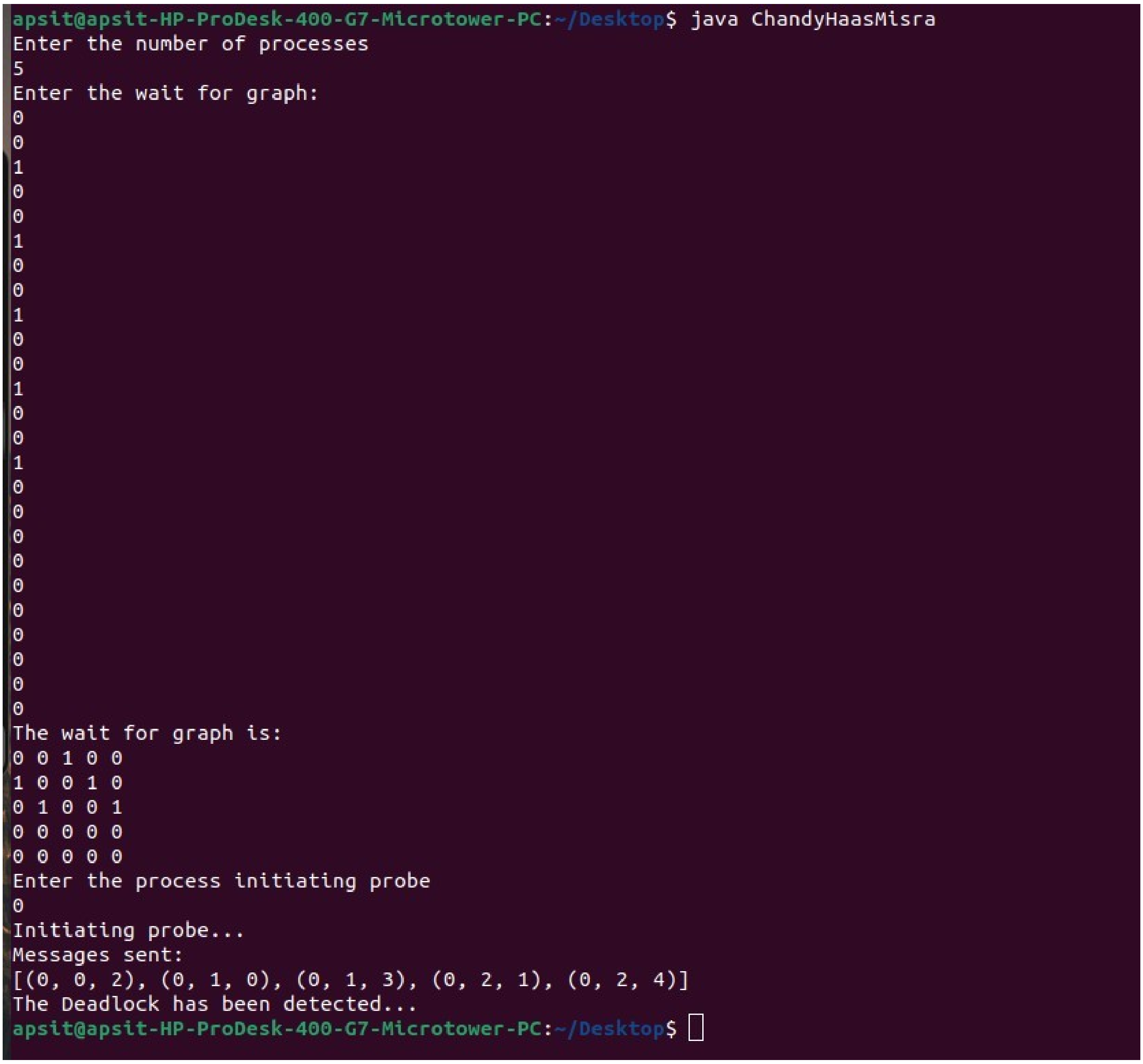
EXPERIMENT 7 SSSS

AIM: Program to demonstrate Chandy-MIisra-Haas distributed deadlock detection algorithm.

CODE:

```
import java.util.*;

class Message {
    public int initiator = 0;
    public int from = 0;
    public int to = 0;

    public Message(int i, int j, int k) {
        initiator = i;
        from = j;
        to = k;
    }

    public String toString() {
        return "(" + initiator + ", " + from + ", " + to + ")";
    }
}

public class ChandyHaasMisra {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int graph[][];
        boolean isDeadlock = false;
        System.out.println("Enter the number of processes");
        int n = sc.nextInt();
        graph = new int[n][n];
        System.out.println("Enter the wait for graph:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                graph[i][j] = sc.nextInt();
            }
        }
        System.out.println("The wait for graph is:");
        new ChandyHaasMisra().display(graph);
        System.out.println("Enter the process initiating probe");
        int init = sc.nextInt();
        System.out.println("Initiating probe...");
        List<Message> mess_list = new ArrayList<Message>();
        int count = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (graph[i][j] == 1) {
                    Message m = new Message(init, i, j);
                    mess_list.add(m);
                    count += 1;
                }
            }
        }
        System.out.println("Messages sent:");
        System.out.println(mess_list);
        for (int i = 0; i < count; i++) {
            for (int j = 0; j < count; j++) {
                if (mess_list.get(i).initiator == mess_list.get(j).to) {
                    isDeadlock = true;
                    break;
                }
            }
            if (isDeadlock) {
                break;
            }
        }
        if (isDeadlock)
            System.out.println("The Deadlock has been detected...");
        else
            System.out.println("No Deadlock has been detected...");
    }

    void display(int[][] mat) {
        int n = mat[0].length;
        int m = mat.length;
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(mat[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:



```
apsit@apsit-HP-ProDesk-400-G7-Microtower-PC:~/Desktop$ java ChandyHaasMisra
Enter the number of processes
5
Enter the wait for graph:
0
0
1
0
0
1
0
0
1
0
0
1
0
0
1
0
0
0
0
0
0
0
0
0
0
The wait for graph is:
0 0 1 0 0
1 0 0 1 0
0 1 0 0 1
0 0 0 0 0
0 0 0 0 0
Enter the process initiating probe
0
Initiating probe...
Messages sent:
[(0, 0, 2), (0, 1, 0), (0, 1, 3), (0, 2, 1), (0, 2, 4)]
The Deadlock has been detected...
apsit@apsit-HP-ProDesk-400-G7-Microtower-PC:~/Desktop$ 
```