Name: Brijesh Rameshbhai Rohit
Admission number : U19CS009

DS-ASSIGNMENT-08

1. Implement Lamport's clock synchronization algorithm and discuss its time complexity.

CODE :

```
// U19CS009
// Brijesh Rohit

#include <iostream>
using namespace std;
void lamportClock(int e1, int e2)
{
   int event[e1][e2];
   int proc1[e1], proc2[e2];

   // Initial timestamps for 1st process
   for (int i = 0; i < e1; i++)
       proc1[i] = i + 1;

   // Initial timestamps for 2nd process
   for (int i = 0; i < e2; i++)
       proc2[i] = i + 1;

   // event[i][j] = 1, if message is sent from ei to ej
   // event[i][j] = -1, if message is received by ei from ej
   // event[i][j] = 0, otherwise

   cout << "\n------- Events of process P1 and P2 interacting
--------\n";
   cout << "\n\t";
```

```
for (int i = 0; i < e2; i++)
{
    cout << "e2" << i + 1 << "\t";
}
for (int i = 0; i < e1; i++)
{
    cout << "\ne1" << i + 1;
    for (int j = 0; j < e2; j++)
    {
        cin >> event[i][j];
    }
}

// updating timestamps
for (int i = 0; i < e1; i++)
{
    for (int j = 0; j < e2; j++)
    {
        if (event[i][j] == 1)
        {
            proc2[j] = max(proc2[j], proc1[i] + 1);
            for (int k = j + 1; k < e2; k++)
                proc2[k] = proc2[k - 1] + 1;
        }
        else if (event[i][j] == -1)
        {
            proc1[i] = max(proc1[i], proc2[j] + 1);
            for (int k = i + 1; k < e1; k++)
                proc1[k] = proc1[k - 1] + 1;
        }
    }
}
```

```
    // display timestamps
    printf("\nTimestamps of events of process P1 : ");
    for (int i = 0; i < e1; i++)
        cout << proc1[i] << " ";
    printf("\nTimestamps of events of process P2 : ");
    for (int j = 0; j < e2; j++)
        cout << proc2[j] << " ";

    cout << "\n";
}

// Lamport Clock Algorithm for 2 process
int main()
{
    unsigned int e1, e2;
    cout << "Enter no of events for process 1 : ";
    cin >> e1;
    cout << "Enter no of events for process 2 : ";
    cin >> e2;
    lamportClock(e1, e2);
    return 0;
}
```

Note:-
event[i][j] = 1, if message is sent from ei to ej
event[i][j] = -1, if message is received by ei from ej
event[i][j] = 0, otherwise

OUTPUT :

```
(base) brijesh@pop-os-birju:~/Documents/ds/ds-assign08$ g++ lamport.cpp
(base) brijesh@pop-os-birju:~/Documents/ds/ds-assign08$ ./a.out
Enter no of events for process 1 : 3
Enter no of events for process 2 : 4

------- Events of process P1 and P2 interacting --------

        e21     e22     e23     e24
e11     1       -1      0       1

e12     -1      -1      0       0

e13     1       0       1       0

Timestamps of events of process P1 : 4 5 6
Timestamps of events of process P2 : 7 8 9 10
```

Time Complexity:-
For 2 process:
No of events in process 1 = E1
No of events in process 2 = E2
Time Complexity = O(E1*E2*(E1+E2))