**C++ program to illustrate the Lamport's**

**// Logical Clock**

#include <bits/stdc++.h>

using namespace std;

// Function to find the maximum timestamp

// between 2 events

int max1(int a, int b)

{

// Return the greatest of th two

if (a > b)

return a;

else

return b;

}

// Function to display the logical timestamp

void display(int e1, int e2,

int p1[5], int p2[3])

{

int i;

cout << "\nThe time stamps of "

"events in P1:\n";

for (i = 0; i < e1; i++) {

cout << p1[i] << " ";

}

cout << "\nThe time stamps of "

"events in P2:\n";

// Print the array p2[]

for (i = 0; i < e2; i++)

cout << p2[i] << " ";

}

// Function to find the timestamp of events

void lamportLogicalClock(int e1, int e2,

int m[5][3])

{

int i, j, k, p1[e1], p2[e2];

// Initialize p1[] and p2[]

for (i = 0; i < e1; i++)

p1[i] = i + 1;

for (i = 0; i < e2; i++)

p2[i] = i + 1;

cout << "\t";

for (i = 0; i < e2; i++)

cout << "\te2" << i + 1;

for (i = 0; i < e1; i++) {

cout << "\n e1" << i + 1 << "\t";

for (j = 0; j < e2; j++)

cout << m[i][j] << "\t";

}

for (i = 0; i < e1; i++) {

for (j = 0; j < e2; j++) {

// Change the timestamp if the

// message is sent

if (m[i][j] == 1) {

p2[j] = max1(p2[j], p1[i] + 1);

for (k = j + 1; k < e2; k++)

p2[k] = p2[k - 1] + 1;

}

// Change the timestamp if the

// message is received

if (m[i][j] == -1) {

p1[i] = max1(p1[i], p2[j] + 1);

for (k = i + 1; k < e1; k++)

p1[k] = p1[k - 1] + 1;

}

}

}

// Function Call

display(e1, e2, p1, p2);

}

// Driver Code

int main()

{

int e1 = 5, e2 = 3, m[5][3];

// message is sent and received

// between two process

/\*dep[i][j] = 1, if message is sent

from ei to ej

dep[i][j] = -1, if message is received

by ei from ej

dep[i][j] = 0, otherwise\*/

m[0][0] = 0;

m[0][1] = 0;

m[0][2] = 0;

m[1][0] = 0;

m[1][1] = 0;

m[1][2] = 1;

m[2][0] = 0;

m[2][1] = 0;

m[2][2] = 0;

m[3][0] = 0;

m[3][1] = 0;

m[3][2] = 0;

m[4][0] = 0;

m[4][1] = -1;

m[4][2] = 0;

// Function Call

lamportLogicalClock(e1, e2, m);

return 0;

