

Name: Brijesh Rohit

Adm. No.: U19CS009

DS-ASSIGNMENT-09

Implement Vector clock algorithm.

CODE=>

```
// U19CS009
// Brijesh Rohit

#include <stdio.h>
#include <stdlib.h>

int max(int a, int b)
{
    if (a >= b)
        return a;
    return b;
}

// VECTOR CLOCK
void vector(int n1, int n2, int message[n1 + 1][n2 + 1])
{
    int p1[n1 + 1][2], p2[n2 + 1][2];
    int i, j;
    // INITIALISING BOTH Processes
    for (i = 0; i <= n1; i++)
    {
        p1[i][0] = i;
        p1[i][1] = 0;
    }

    for (i = 0; i <= n2; i++)
    {
        p2[i][0] = 0;
        p2[i][1] = i;
    }

    //CHECK FOR MESSAGE TRANSFER
    for (i = 1; i <= n1; i++)
    {
        for (j = 1; j <= n2; j++)
        {
            // IF MESSAGE SENT FROM P1 TO P2
```

```

// THEN THE VECTOR OF P2 WOULD UPDATE
// MAXIMUM OF THE TIMESTAMPS.
if (message[i][j] == 1)
{
    p2[j][0] = max(p2[j][0], p1[i][0]);
    p2[j][1] = max(p2[j][1], p1[i][1]);

    int k;
    // UPDATING THE REST OF THE VALUES
    for (k = j + 1; k <= n2; k++)
    {
        p2[k][0] = p2[k - 1][0];
        p2[k][1] = p2[k - 1][1] + 1;
    }
}
// IF MESSAGE SENT FROM P2 TO P1
// THEN THE VECTOR OF P1 WOULD UPDATE
// MAXIMUM OF THE TIMESTAMPS.
else if (message[i][j] == -1)
{
    p1[i][0] = max(p2[j][0], p1[i][0]);
    p1[i][1] = max(p2[j][1], p1[i][1]);
    int k;
    // UPDATING THE REST OF THE VALUES
    for (k = i + 1; k <= n1; k++)
    {
        p1[k][0] = p1[k - 1][0] + 1;
        p1[k][1] = p1[k - 1][1];
    }
}
}
}
// DISPLAYING THE VALUE.
printf("Process 1\tProcess2\n");
for (int i = 0; i <= max(n1, n2); i++)
{
    if (i <= n1)
    {
        printf("[ %d , %d ]\t", p1[i][0], p1[i][1]);
    }
    else
    {
        printf("\t\t");
    }
}

if (i <= n2)

```

```

        {
            printf("[ %d , %d ]\t", p2[i][0], p2[i][1]);
        }
        printf("\n");
    }
}

int main()
{
    int n1 = 8, n2 = 13;
    int message[n1 + 1][n2 + 1];
    int i, j;
    for (i = 1; i <= n1; i++)
        for (j = 1; j <= n2; j++)
            message[i][j] = 0;

    message[2][5] = 1; // P1->P2
    message[4][4] = -1; // P2->P1
    message[6][8] = 1; // P1->P2
    message[9][9] = -1; // P2->P1
    vector(n1, n2, message);

    return 0;
}

```

OUTPUT=>

```
PS C:\Users\brijesh\Documents\ds\ds-assign09> cd "c:\Users\brijesh\Do
-o vector } ; if ($?) { .\vector }
Process 1      Process2
[ 0 , 0 ]      [ 0 , 0 ]
[ 1 , 0 ]      [ 0 , 1 ]
[ 2 , 0 ]      [ 0 , 2 ]
[ 3 , 0 ]      [ 0 , 3 ]
[ 4 , 4 ]      [ 0 , 4 ]
[ 5 , 4 ]      [ 2 , 5 ]
[ 6 , 4 ]      [ 2 , 6 ]
[ 7 , 4 ]      [ 2 , 7 ]
[ 8 , 4 ]      [ 6 , 8 ]
               [ 6 , 9 ]
               [ 6 , 10 ]
               [ 6 , 11 ]
               [ 6 , 12 ]
               [ 6 , 13 ]
PS C:\Users\brijesh\Documents\ds\ds-assign09> 
```