MICROSOFT PREPARATION

DAY : 01

Number of Connected Computers :

Problem Link :
https://www.codingninjas.com/studio/problems/number-of-connected-computers_1281389?

Test Cases Passed : 10 / 11

Time Used : 27.14 min

Difficulty Level : MEDIUM

Approach Used :

totalConnectedComp( )
- Declare visited vector of same size of grid initially assigned 0 to all elements
- Traverse through all components and if its a computer and not visited then :
  - Make a DFS call to mark it and its adjacent computers as dfs(row,col,graph,visited,rowsize,colsizel,connections)
- Traverse through the visited vector and count the number of 1s in the vector
- Return count of 1s

dfs( )
- Mark the given node as visited
- Traverse through the complete row and check if there is a computer unvisited :
  - Increase the connection count
  - Make a dfs call as dfs(row,colsize,graph,visited,connections)
- Traverse through the complete row and check if there is a computer unvisited :
  - Increase the connection count
  - Make a dfs call as dfs(rowsize,col,graph,visited,connections)
- Check if connections are zero :
  - If zero then un-visit the node element

Solution :

```
#include <bits/stdc++.h>
```

```
bool dfs(int counter,int row,int
col,vector<vector<int>>&visited,vector<vector<int>>&arr,int n,int m)
{
      // mark the node as visited
      visited[row][col] = 1;
    // traverse through the row
      for(int i=0;i<m;i++)
      {
        // check if not visited and is a computer
            if(!visited[row][i] && arr[row][i]==1)
            {
            // increase counter and call dfs
                  counter+=1;
                  dfs(counter,row,i,visited,arr,n,m);
            }
      }
    // traverse through all columns
      for(int i=0;i<n;i++)
      {
        // check if not visited and is a computer
            if(!visited[i][col] && arr[i][col]==1)
            {
            // increase counter and call dfs
                  counter+=1;
                  dfs(counter,i,col,visited,arr,n,m);
            }
      }
    // check if counter is greater than 1
      if(counter<1)
      {
        // if not greater than 1 then unvisit the computer
            visited[row][col]=0;
      }
}
int totalConnectedComp(vector < vector < int > > &arr, int n, int m) {
      // Declare a visited vector of same size as the arr
      vector<vector<int>> visited(n,vector<int>(m,0));
    // create a counter for connected computers
      int counter = 0;
    // traverse through all the components
      for(int i=0;i<n;i++)
      {
            for(int j=0;j<m;j++)
```

```
            {
            // check if component is a computer and is unvisited
                  if(!visited[i][j] && arr[i][j]==1)
                  {
                // call dfs to visit the component
                        dfs(0,i,j,visited,arr,n,m);
                  }
            }
      }
      // traverse through all the visited components
      for(int i=0;i<n;i++)
      {
            for(int j=0;j<m;j++)
            {
            // if a component is visited and is a computer then increase
connected computer count
                  if(visited[i][j]==1 && arr[i][j]==1)
                  {
                        counter++;
                  }
            }
      }
    // return connected computer count
      return counter;
}
```