**Slip 18**

Q.1) Write a program to implement Graph Coloring Algorithm

#include<iostream>

#define NODE 6

using namespace std;

int graph[NODE][NODE] = {

   {0, 1, 1, 1, 0, 0},

   {1, 0, 0, 1, 1, 0},

   {1, 0, 0, 1, 0, 1},

   {1, 1, 1, 0, 1, 1},

   {0, 1, 0, 1, 0, 1},

   {0, 0, 1, 1, 1, 0}

};

void graphColoring() {

   int color[NODE];

   color[0] = 0;    //Assign first color for the first node

   bool colorUsed[NODE];    //Used to check whether color is used or not

   for(int i = 1; i<NODE; i++)

      color[i] = -1;    //initialize all other vertices are unassigned

   for(int i = 0; i<NODE; i++)

      colorUsed[i] = false;    //initially any colors are not chosen

   for(int u = 1; u<NODE; u++) {    //for all other NODE - 1 vertices

      for(int v = 0; v<NODE; v++) {

         if(graph[u][v]){

            if(color[v] != -1)    //when one color is assigned, make it unavailable

               colorUsed[color[v]] = true;

         }

     }

     int col;

     for(col = 0; col<NODE; col++)

        if(!colorUsed[col])    //find a color which is not assigned

           break;

     color[u] = col;    //assign found color in the list

     for(int v = 0; v<NODE; v++) {    //for next iteration make color availability to false

        if(graph[u][v]) {

           if(color[v] != -1)

              colorUsed[color[v]] = false;

        }

     }

  }

  for(int u = 0; u<NODE; u++)

     cout <<"Color: " << u << ", Assigned with Color: " <<color[u] <<endl;

}

main() {

   graphColoring();

}