

Name: Vanessa D'mello
Roll No. 8863
Branch: SE Computers A (Batch A)
Experiment 15: Graph Colouring

```
#include<stdio.h>
int n,sol[10]={0},count=0,adj[11][11]={0},m;
int can_color(int node,int c)
{
    int j;
    for(j=1;j<=node-1;j++)
    {
        if(adj[j][node]==1 && sol[j]==c)//check whether the adjacent node's colour is same
as c
        return 0;
    }
    return 1;
}
void m_color(int node)
{
    int i,j;
    for(i=1;i<=m;i++)
    {
        if(can_color(node,i))//check whether the node can be coloured with i
        {
            sol[node]=i;
            if(node==n) //print solution vector
            {
                count++;
                printf("\n***** Solution Vector : %d *****",count);
                printf("\n");
                for(j=1;j<=n;j++)
                {
                    if(sol[j]==1) //Changing 1 to Red
                        printf("RED\t");
                    else if(sol[j]==2) //Changing 2 to Green
                        printf("GREEN\t");
                    else if(sol[j]==3) //Changing 3 to Blue
                        printf("BLUE\t");
                    else if(sol[j]==4) //Changing 4 to Yellow
                        printf("YELLOW\t");
                    else if(sol[j]==5) //Changing 5 to Violet
                        printf("VIOLET\t");
                }
                printf("\n");
            }
            else
                m_color(node+1); //Call to m_color method
        }
    }
}
```

```

int main()
{
    int i,s,e,d,deg;
    printf("Enter the degree of the graph : "); //degree of the graph
    scanf("%d", &deg);
    m=deg+1;
    printf("Enter no of vertices : ");
    scanf("%d", &n);
    printf("Enter no of edges : ");
    scanf("%d", &e);
    for (i = 1; i <= e; i++) //enter the graph
    {

        printf("Enter source node :");
        scanf("%d", &s);
        if (s > n || s == 0)
        {
            printf("Invalid source Node\nEnter again ");
            i--;
            continue;
        }
        printf("Enter destination node :");
        scanf("%d", &d);
        if (d > n || s == 0)
        {
            printf("Invalid destination Node\nEnter again ");
            i--;
            continue;
        }
        adj[s][d] = 1; //Undirected Graph
        adj[d][s] = 1;
    }
    m_color(1);
}

```

Output:

C:\Users\dmell\OneDrive\Desktop\Subjects\AOA\GraphColouring

```
Enter the degree of the graph : 2
Enter no of vertices : 5
Enter no of edges : 7
Enter source node :1
Enter destination node :2
Enter source node :1
Enter destination node :3
Enter source node :2
Enter destination node :4
Enter source node :2
Enter destination node :5
Enter source node :3
Enter destination node :4
Enter source node :3
Enter destination node :5
Enter source node :4
Enter destination node :5

***** Solution Vector : 1 *****
RED    GREEN  GREEN  RED    BLUE

***** Solution Vector : 2 *****
RED    GREEN  GREEN  BLUE   RED

***** Solution Vector : 3 *****
RED    BLUE   BLUE   RED    GREEN

***** Solution Vector : 4 *****
RED    BLUE   BLUE   GREEN  RED

***** Solution Vector : 5 *****
GREEN  RED    RED    GREEN  BLUE

***** Solution Vector : 6 *****
GREEN  RED    RED    BLUE   GREEN
```