Graph Coloring Problem

```
#include<stdio.h>
int G[50][50],x[50];
void next_color(int k){
  int i,j;
  x[k]=1;
  for(i=0;i<k;i++){
    if(G[i][k]!=0 && x[k]==x[i])
      x[k]=x[i]+1;
  }
}

int main(){
 int n,e,i,j,k,l;
 printf("Enter no. of vertices : ");
 scanf("%d",&n);
 printf("Enter no. of edges : ");
 scanf("%d",&e);

 for(i=0;i<n;i++)
   for(j=0;j<n;j++)
     G[i][j]=0;

 printf("Enter indexes where value is 1-->\n");
 for(i=0;i<e;i++){
   scanf("%d %d",&k,&l);
   G[k][l]=1;
```

```
  G[l][k]=1;
 }

 for(i=0;i<n;i++)
  next_color(i);

 printf("Colors of vertices -->\n");
 for(i=0;i<n;i++)
  printf("Vertex[%d] : %d\n",i+1,x[i]);

 return 0;
```

Sample input:

```
Enter no. of vertices : 10
Enter no. of edges : 15
Enter indexes where value is 1-->
1 2
1 7
1 9
2 5
2 7
2 10
3 5
3 7
4 5
4 6
4 9
```

5 6

5 10

8 9

8 10

Output:

Colors of vertices -->

Vertex[1] : 1

Vertex[2] : 1

Vertex[3] : 2

Vertex[4] : 1

Vertex[5] : 1

Vertex[6] : 2

Vertex[7] : 3

Vertex[8] : 3

Vertex[9] : 1

Vertex[10] : 2

Process returned 0 (0x0) execution time : 5.928 s

Press any key to continue.

N- Queen Problem

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

using namespace std;
int a[30],count=0;
///row=index & column = value
int place(int pos)
{
   int i;
   for(i=1;i<pos; i++)
   {
      if((a[i]==a[pos])||(abs(a[i]-a[pos])==abs(i-pos)))
      {
         return 0;
      }

   }
   return 1;
}
void print_sol(int n)
{
   int i,j;
   count++;
   cout<<"\n\n Solution # "<<endl;
   for(i=1; i<=n; i++)
   {
      for(j=1; j<=n; j++)
```

```
        {
            if(a[i]==j)
            {
                cout<<"Q\t";
            }
            else
            {
                cout<<"*\t";
            }

        }
        cout<<endl;
    }
}
void Queen(int n)
{
    int k=1;
    a[k]=0;
    while(k!=0)
    {
        do
        {
            a[k]++;
        }
        while((a[k]<=n)&&!place(k));
        if(a[k]<=n)
        {
            if(k==n)
            {
```

```
            print_sol(n);
          }
          else
          {
            k++;
            a[k]=0;

          }
        }
        else
        {
          k--;
        }
    }
}
int main()
{
    int i,n;
    cout<<"Enter number of queens"<<endl;
    cin>>n;
    Queen(n);
    cout<<"Total Solutions are : "<< count<<endl;
    return 0;
}
```

Sample Input:

Enter number of queens:

Output:

Solution #

Q * * * * * * *
* * * * Q * * *
* * * * * * * Q
* * * * * Q * *
* * Q * * * * *
* * * * * * Q *
* Q * * * * * *
* * * Q * * * *

Solution #

Q * * * * * * *
* * * * * Q * *
* * * * * * * Q
* * Q * * * * *
* * * * * * Q *
* * * Q * * * *
* Q * * * * * *
* * * * Q * * *

Solution #

Q * * * * * * *
* * * * * * Q *

```
*  *  *  Q  *  *  *  *
*  *  *  *  *  Q  *  *
*  *  *  *  *  *  *  Q
*  Q  *  *  *  *  *  *
*  *  *  *  Q  *  *  *
*  *  Q  *  *  *  *  *
```

Solution #

```
Q  *  *  *  *  *  *  *
*  *  *  *  *  *  Q  *
*  *  *  *  Q  *  *  *
*  *  *  *  *  *  *  Q
*  Q  *  *  *  *  *  *
*  *  *  Q  *  *  *  *
*  *  *  *  *  Q  *  *
*  *  Q  *  *  *  *  *
```

……

……

……..

……

……

……

……..

……

Process returned 0 (0x0) execution time : 5.928 s

Press any key to continue.