**10.N queens**

#include<stdio.h>

#include<conio.h>

#include<math.h>

int a[30],count=0;

//row=index and 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++;

printf("\n\nSolution #%d:\n",count);

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

if(a[i]==j)

printf("Q\t");

else

printf("\*\t");

}

printf("\n");

}

}

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--;

}

}

void main()

{

int i,n;

printf("Enter the number of Queens\n");

scanf("%d",&n);

queen(n);

printf("\nTotal solutions=%d",count);

}

**7.Floyds**

**#include<stdio.h>**

**#include<stdlib.h>**

**int cost[10][10], a[10][10];**

**int min(int a, int b)**

**{**

**return ((a < b) ? a : b);**

**}**

**void allpaths(int cost[10][10], int a[10][10], int n)**

**{**

**int i, j, k;**

**for(i = 1; i <= n; i++){**

**for(j = 1; j <= n; j++)**

**a[i][j] = cost[i][j];**

**}**

**for(k = 1; k <= n; k++){**

**for(i = 1; i <= n; i++){**

**for(j = 1; j <= n; j++){**

**a[i][j] = min(a[i][j], a[i][k] + a[k][j]);**

**}**

**}**

**}**

**}**

**int main()**

**{**

**int i, j, n;**

**printf("\n\nEnter the no. of the vertices: ");**

**scanf("%d", &n);**

**printf("\nEnter the cost adjacency matrix: \n");**

**for(i = 1; i <= n; i++)**

**for(j = 1; j<= n; j++)**

**scanf("%d", &cost[i][j]);**

**allpaths(cost, a, n);**

**printf("The shortest path obtained is as follows: \n");**

**for(i = 1; i <= n; i++){**

**for(j = 1; j <= n; j++)**

**printf("%d\t", a[i][j]);**

**printf("\n");**

**}**

**return 0;**

**}**

**9.Dijkstra**

**#include<stdio.h>**

**#include<stdlib.h>**

**void dij(int n,int cost[10][10],int s,int dist[10])**

**{**

**int i,v,count=1,min,visited[10];**

**for(i=1;i<=n;i++){**

**visited[i]=0;**

**dist[i]=cost[s][i];**

**}**

**visited[s]=1;**

**dist[s]=0;**

**while(count<=n){**

**min=999;**

**for(i=1;i<=n;i++)**

**if(dist[i]<min && visited[i]==0){**

**min=dist[i];**

**v=i;**

**}**

**visited[v]=1;**

**count++;**

**for(i=1;i<=n;i++)**

**if(dist[i]>dist[v]+cost[v][i])**

**dist[i]=cost[v][i];**

**}**

**}**

**int main(){**

**int i,j,n,s,cost[10][10],dist[10];**

**printf("\n Enter number of nodes\n");**

**scanf("%d",&n);**

**printf("\nEnter cost matrix\n");**

**for(i=1;i<=n;i++){**

**for(j=1;j<=n;j++){**

**scanf("%d",&cost[i][j]);**

**}**

**}**

**printf("\nEnter source\n");**

**scanf("%d",&s);**

**dij(n,cost,s,dist);**

**printf("\n Shortest path from %d is\n",s);**

**for(i=1;i<=n;i++)**

**if(s!=i)**

**printf("%d->%d=%d",s,i,dist[i]);**

**return 0;**

**}**