**queen**

#include<iostream>

#include<bits/stdc++.h>

using namespace std;

int a[30],count=0;

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;

count++;

cout<<"solution"<<count<<endl;

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

{

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

{

if(a[i]==a[j])

{

cout<<"Q ";

}

else

{

cout<<"\* ";

}

}

cout<<endl;

}

}

void queen(int n)

{

int k=1;

a[k]=0;

while(k!=0)

{

a[k]=a[k]+1;

while((a[k]<=n)&& !place(k))

a[k]++;

if(a[k]<=n)

{

if(k==n)

{

print\_sol(n);

}

else

{

k++;

a[k]=0;

}

}

else

{

k--;

}

}

}

int main()

{

int i,n;

cout<<"no. of queens"<<endl;

cin>>n;

queen(n);

//cout<<"solution total:"<<count<<endl;

}

**Tsm:**

#include<bits/stdc++.h>

#include<iostream>

using namespace std;

int a[10][10], visited[10], n, cost = 0;

void get() {

int i, j;

cout<<"Enter No. of Cities:"<<endl;

cin>>n;

cout<<"\nEnter Cost Matrix: \n"<<endl;

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

cout<<"\n Enter Elements of Row# :\n"<<i+1;

for (j = 0; j < n; j++)

cin>>a[i][j];

visited[i] = 0;

}

cout<<"\nThe cost list is:\n";

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

cout<<"\n\n";

for (j = 0; j < n; j++)

cout<<"\t"<<a[i][j];

}

}

int least(int c) {

int i, nc = 999;

int min = 999, kmin;

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

if ((a[c][i] != 0) && (visited[i] == 0))

if (a[c][i] < min) {

min = a[i][0] + a[c][i];

kmin = a[c][i];

nc = i;

}

}

if (min != 999)

cost += kmin;

return nc;

}

void mincost(int city) {

int i, ncity;

visited[city] = 1;

cout<<city+1<<"->";

ncity = least(city);

if (ncity == 999) {

ncity = 0;

cout<<ncity+1;

cost += a[city][ncity];

return;

}

mincost(ncity);

}

void put() {

cout<<"Minimum cost:"<<endl;

cout<<cost;

}

int main() {

get();

cout<<"\nThe Path is:\n";

mincost(0);

put();

}



