EXPT 8:

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#define MAX 20

char arr[MAX][MAX];

void print\_grid(int n,int x[])

{

int i,j;

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

{

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

{

arr[i][j]='-';

}

}

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

{

arr[i][x[i]]='Q';

}

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

{

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

{

printf("\t%c",arr[i][j]);

}

printf("\n");

}

}

//for checking queens placement is safe or not

int safetoplace(int x[],int k)

{

int i;

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

{

if(x[i]==x[k]||i-x[i]==k-x[k]||i+x[i]==k+x[k])

{

return 0;//false

}

}

return 1;//true

}

//for printing queens and placing them in grid

void nqueens(int n)

{

int x[20];

int count=0;

int k=1;

x[k]=0;

while(k!=0)

{

x[k]=x[k]+1;

while((x[k]<=n)&&(!safetoplace(x,k)))

{

x[k]=x[k]+1;

}

if(x[k]<=n)

{

if(k==n)

{

count++;

printf("\n\tPlacement %d is : \n\n\n",count);

print\_grid(n,x);

getch();

}

else

{

k++;

x[k]=0;

}

}

else

{

k--;

}

}

return;

}

int main()

{

int n;

printf("\t C PROGRAM OF N-QUEEN PROBLEM\n\n");

printf("\nEnter the no. of Queens :");

scanf("%d",&n);

printf("\n\nUSING %D QUEEN'S STRATEGY \n\n",n);

nqueens(n);

system("PAUSE");

return 0;

}