**N Queens Problem in C Using Backtracking**

Here you will get program for N queens problem in C using backtracking.

N Queens Problem is a famous puzzle in which n-queens are to be placed on a nxn chess board such that no two queens are in the same row, column or diagonal. In this tutorial I am sharing the C program to find solution for N Queens problem using backtracking. Below animation shows the solution for 8 queens problem using backtracking.

#include<stdio.h>

#include<math.h>

int board[20],count;

int main()

{

int n,i,j;

void queen(int row,int n);

printf(" - N Queens Problem Using Backtracking -");

printf("\n\nEnter number of Queens:");

scanf("%d",&n);

queen(1,n);

return 0;

}

//function for printing the solution

void print(int n)

{

int i,j;

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

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

  printf("\t%d",i);

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

{

  printf("\n\n%d",i);

  for(j=1;j<=n;++j) //for nxn board

  {

   if(board[i]==j)

    printf("\tQ"); //queen at i,j position

   else

    printf("\t-"); //empty slot

  }

}

}

/\*funtion to check conflicts

If no conflict for desired postion returns 1 otherwise returns 0\*/

int place(int row,int column)

{

int i;

for(i=1;i<=row-1;++i)

{

  //checking column and digonal conflicts

  if(board[i]==column)

   return 0;

  else

   if(abs(board[i]-column)==abs(i-row))

    return 0;

}

return 1; //no conflicts

}

//function to check for proper positioning of queen

void queen(int row,int n)

{

int column;

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

{

  if(place(row,column))

  {

   board[row]=column; //no conflicts so place queen

   if(row==n) //dead end

    print(n); //printing the board configuration

   else //try queen with next position

    queen(row+1,n);

  }

}

}