**PRACTICAL-6**

**AIM:** Write a program to implement Single Player Game .(Using Heuristic Function)

**Source Code:**

e#include <stdio.h>

//Number of queens

int N;

//chessboard

int board[100][100];

//function to check if the cell is attacked or not

int is\_attack(int i,int j)

{

int k,l;

//checking if there is a queen in row or column

for(k=0;k<N;k++)

{

if((board[i][k] == 1) || (board[k][j] == 1))

return 1;

}

//checking for diagonals

for(k=0;k<N;k++)

{

for(l=0;l<N;l++)

{

if(((k+l) == (i+j)) || ((k-l) == (i-j)))

{

if(board[k][l] == 1)

return 1;

}

}

}

return 0;

}

int N\_queen(int n)

{

int i,j;

//if n is 0, solution found

if(n==0)

return 1;

for(i=0;i<N;i++)

{

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

{

//checking if we can place a queen here or not

//queen will not be placed if the place is being attacked

//or already occupied

if((!is\_attack(i,j)) && (board[i][j]!=1))

{

board[i][j] = 1;

//recursion

//wether we can put the next queen with this arrangment or not

if(N\_queen(n-1)==1)

{

return 1;

}

board[i][j] = 0;

}

}

}

return 0;

}

int main()

{

//taking the value of N

printf("Enter the value of N for NxN chessboard\n");

scanf("%d",&N);

int i,j;

//setting all elements to 0

for(i=0;i<N;i++)

{

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

{

board[i][j]=0;

}

}

//calling the function

N\_queen(N);

//printing the matix

for(i=0;i<N;i++)

{

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

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

printf("\n");

}

}

**Output:**

