

12) Design and implement C/C++ Program for N Queen's problem using Backtracking. 12 Design and implement C/C++ Program for N Queen's problem using Backtracking.

Sol:

```
#include<stdio.h>

#include<conio.h>

void nqueens(int);

int place(int[],int);

void prins(int n,int x[])

{

    char c[10][10];

    int i,j;

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

    {

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

        {

            c[i][j]='x';

        }

    }

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

    {

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

    }

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

    {

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

```

        {

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

        }

        printf("\n");

    }

}

void main()

{

    int n;

    printf("enter the number of queens:\t");

    scanf("%d",&n);

    if(n==2 || n==3)

    {

        printf("no solution for %d queens\n",n);

    }

    else{

        nqueens(n);

    }

}

void nqueens(int n)

{

    int k,x[10],count=0;

    k=1;

    x[k]=0;

    while(k!=0)

    {

        x[k]++;

```

```

while(place(x,k)==1 && x[k]<=n)

{

    x[k]++;

}

if(x[k]<=n)

{

    if(k==n)

    {

        printf("\n solution %d is\n",++count);

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

        {

            printf("%d---->%d is \n",k,x[k]);

        }

        printf("solution in the form of chess board\n");

        prins(n,x);

    }

    else

    {

        k++;

        x[k]=0;

    }

}

else{

    k--;

}

}

}

```

```
int place(int x[],int k)
{
    int i;
    for(i=1;i<=k-1;i++)
    {
        if(i-x[i]==k-x[k] || i+x[i]==k+x[k] || x[i]==x[k])
        {
            return 1;
        }
    }
    return 0;
}
```

OUTPUT:

enter the number of queens: 4

solution 1 is

1---->2 is

2---->4 is

3---->1 is

4---->3 is

solution in the form of chess board

x	Q	x	x
---	---	---	---

x	x	x	Q
---	---	---	---

Q	x	x	x
---	---	---	---

x	x	Q	x
---	---	---	---

solution 2 is

1---->3 is

2---->1 is

3---->4 is

4---->2 is

solution in the form of chess board

x	x	Q	x
---	---	---	---

Q	x	x	x
---	---	---	---

x	x	x	Q
---	---	---	---

x	Q	x	x
---	---	---	---