```c
#include<stdio.h>
#include<stdlib.h>
#define queen 0
#define row 1
#define col 2
#define nwtose 3
#define swtone 4

struct board* Board=NULL;

struct data{
    int* d;
};

struct board{
    struct data* head;
};

struct board* intialize(int n){
    struct board* temp=(struct board*)malloc(sizeof(struct board));
    temp->head=(struct data*)malloc(5*sizeof(struct data));
    for(int i=0;i<3;++i)
        temp->head[i].d=(int*)malloc(n*sizeof(int));
    for(int i=3;i<5;++i)
        temp->head[i].d=(int*)malloc((2*n-1)*sizeof(int));

   for(int i=0;i<n;++i){
    temp->head[queen].d[i]=-1;
   temp->head[row].d[i]=temp->head[col].d[i]=0;
   }

    for(int i=0;i<2*n-1;++i)
    temp->head[nwtose].d[i]=temp->head[swtone].d[i]=0;

  return temp;
}

bool free(int i,int j,int n){
    return(Board->head[1].d[i]==0&&Board->head[2].d[j]==0 && Board->head[3].d[j-i+n-1]==0&&Board->head[4].d[j+i]==0);
}

void addqueen(int i,int j,int n){
    Board->head[queen].d[i]=j;
    Board->head[row].d[i]=1;
    Board->head[col].d[j]=1;
    Board->head[nwtose].d[j-i+n-1]=1;
    Board->head[swtone].d[j+i]=1;
}

void undoqueen(int i,int j,int n){
   Board->head[queen].d[i]=-1;
    Board->head[row].d[i]=0;
    Board->head[col].d[j]=0;
```

```
    Board->head[nwtose].d[j-i+n-1]=0;
    Board->head[swtone].d[j+i]=0;
}

void printsol(int n){
    printf("_____\n");
 int* d;int i=

      for(int j=0;j<n;++j)
        {  printf("| ");
          if(Board->head[queen].d[i]==j)
           printf(" Q ");
          else
           printf("   ");
       }
       printf("|");
       printf("\n|____|____|____|____|____|____|____|____|");
    printf("\n");
}
printf("\n-------------------------------------------------\n");
}
bool placequeen(int i,int n){
      bool extendsoln=false,check=false;
         for(int j=0;j<n;++j){
            if(free(i,j,n)){
             addqueen(i,j,n);
             if(i==n-1)
              printsol(n);
             else
              extendsoln=placequeen(i+1,n);
             if(extendsoln){
             check=true;
              return true;
            }
            else
              undoqueen(i,j,n);
         }
       }
       if(check==false)return false;
}


int main(){
    int n;
    printf("Enter the number of queen:: ");
    sca(Board->head[1].d[i]==0&&Board->head[2].d[j]==0 && Board->head[3].d[j-i+n-1]==0
    if(n==2||n==3){
    printf("No solution exist");
   return 0;
}
    Board=intialize(n);
    if(placequeen(0,n))
      //printsol(n);

```

```
108:        return 0;
109: }
```