```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h>


#define ROW 7
#define COLUMN 7
#define ROUNDS 10
#define TRUE 1
#define FALSE 0

// 42 = *
// 63 = ?

void initializeBoards(int**,int**,int);
int randomDiamond();
int getGuessedCoordinates(int,int);
int checkDiamonds(int,int,int**);
void provideHints(int,int,int**,int**);
void displayUserBoard(int**);
void displayDiamondBoard(int **);

int main()
{

srand(time(NULL));
int **userBoard, **diamondBoard;
int i,n;//number of diamonds
int rowCoordinate,columnCoordinate;
int totalPoint=0,bestScore=0;
char playAgain;

userBoard=(int**)malloc(ROW*sizeof(int*));
diamondBoard=(int**)malloc(ROW*sizeof(int*));

if(userBoard==NULL || diamondBoard==NULL)
{
    printf("Out of memory!");
    return 0;
}
for(i=0;i<ROW;i++)
{
    userBoard[i]=(int*)malloc(COLUMN*sizeof(int));
    diamondBoard[i]=(int*)malloc(COLUMN*sizeof(int));
    if(userBoard[i]==NULL || diamondBoard[i]==NULL)
    {
        printf("Out of memory!");
        return 0;
    }
}

int roundCounter=1;
int diamondCounter=0;


while(roundCounter<=ROUNDS)
{
    if(roundCounter==1)
    {
        printf("\n*Diamond-Hunter*\nLets get started!\n");
        printf("Enter the number of diamonds to hunt:");
        scanf("%d",&n);
        while(n>49 || n<1)
        {
            printf("please enter number of diamonds between 1-49:");
            scanf("%d",&n);
```

```
        }
        initializeBoards(userBoard,diamondBoard,n);
        displayUserBoard(userBoard);
    }

    printf("\n\n");
    printf("Round %d:\n",roundCounter);
    printf("Enter the coordinates of the diamonds: ");
    scanf("%d %d",&rowCoordinate,&columnCoordinate);
    fflush(stdin);

    int checkGuessedCoordinates=getGuessedCoordinates(rowCoordinate,columnCoordinate);
    while(checkGuessedCoordinates==0)
    {
        printf("Please enter coordinates between 1-7! ");
        scanf("%d %d",&rowCoordinate,&columnCoordinate);
        fflush(stdin);
        checkGuessedCoordinates=getGuessedCoordinates(rowCoordinate,columnCoordinate);
    }

    int checkDia=checkDiamonds(rowCoordinate,columnCoordinate,diamondBoard);
    if(checkDia==1)
    {
        printf("You got 100 points!\n");
        totalPoint+=100;
        diamondCounter++;
    }
    else
    {
        printf("You got 0 points!\n");
    }

    provideHints(rowCoordinate,columnCoordinate,userBoard,diamondBoard);
    printf("Your total points is %d!\n",totalPoint);
    displayUserBoard(userBoard);
    if(diamondCounter==n || roundCounter==ROUNDS)
    {
        if(totalPoint>bestScore)
            bestScore=totalPoint;
        if(diamondCounter==n)
            printf("\nYou won! You found all diamons!\n");
        else
            printf("\nYou run out of rounds! Game over!\n");

        printf("Diamond Board is as follows:\n");
        displayDiamondBoard(diamondBoard);
        printf("\nYour score is %d\n",totalPoint);
        printf("Your best score is %d\n",bestScore);
        printf("Do you want to play again?(Y/N)");
        scanf("%c",&playAgain);
        fflush(stdin);
        while(playAgain!='Y' && playAgain!='N')
        {
            printf("Please enter 'Y' for YES, 'N' for NO!\n");
            printf("Do you want to play again?(Y/N)");
            scanf("%c",&playAgain);
            fflush(stdin);
        }
        if(playAgain=='Y')
        {
            roundCounter=0;
            totalPoint=0;
            diamondCounter=0;
        }
        else
        {
```

```
            printf("\nSee you!\n");
            break;
        }

    }
    roundCounter++;
}

    return 0;
}

void initializeBoards(int** userBoard,int** diamondBoard,int n)
{
    int i,j,k;
    int coordinate1,coordinate2;

    for(i=0;i<ROW;i++)
    {
        for(j=0;j<COLUMN;j++)
        {
            userBoard[i][j]=63;
        }
    }
    for(i=0;i<ROW;i++)
    {
        for(j=0;j<COLUMN;j++)
        {
            diamondBoard[i][j]=63;
        }
    }

    for(k=0;k<n;k++)
    {
        coordinate1=randomDiamond();
        coordinate2=randomDiamond();

        for(i=0;i<ROW;i++)
        {
            for(j=0;j<COLUMN;j++)
            {
                if(i==coordinate1 && j==coordinate2 && diamondBoard[i][j]!='*')
                    diamondBoard[i][j]=42;
                else if(i==coordinate1 && j==coordinate2 && diamondBoard[i][j]=='*')
                {
                    k--;
                }

            }
        }
    }




}

int randomDiamond()
{
    return rand()%7;
}

int getGuessedCoordinates(int row,int column)
{
    if(row<1 || row>7 || column<1 || column>7)
        return FALSE;
```

```
    else
        return TRUE;
}

int checkDiamonds(int row, int column, int **diamondBoard)
{

    if(diamondBoard[row-1][column-1]=='*')
        return TRUE;
    else
        return FALSE;
}

void provideHints(int row, int column, int** userBoard, int** diamondBoard)
{

    int i,j;
    int diaCounter=0;

    if(diamondBoard[row-1][column-1]=='*')
    {
        userBoard[row-1][column-1]='*';
    }
    else if(row==1 && column==1)
    {
        if(diamondBoard[row][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row][column]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;

    }
    else if(row==7 && column==1)
    {
        if(diamondBoard[row-2][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(row==1 && column==7)
    {
        if(diamondBoard[row-1][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row][column-1]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(row==7 && column==7)
    {
        if(diamondBoard[row-2][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column-2]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }

    else if(row==1 && column>=2 && column<=6)
```

```
    {
            if(diamondBoard[row-1][column-2]=='*')
                diaCounter++;
            if(diamondBoard[row][column-2]=='*')
                diaCounter++;
            if(diamondBoard[row][column-1]=='*')
                diaCounter++;
            if(diamondBoard[row][column]=='*')
                diaCounter++;
            if(diamondBoard[row-1][column]=='*')
                diaCounter++;
            userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(column==1 && row>=2 && row<=6)
    {
        if(diamondBoard[row-2][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column]=='*')
            diaCounter++;
        if(diamondBoard[row][column]=='*')
            diaCounter++;
        if(diamondBoard[row][column-1]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(row==7 && column>=2 && column<=6)
    {
            if(diamondBoard[row-1][column-2]=='*')
                diaCounter++;
            if(diamondBoard[row-2][column-2]=='*')
                diaCounter++;
            if(diamondBoard[row-2][column-1]=='*')
                diaCounter++;
            if(diamondBoard[row-2][column]=='*')
                diaCounter++;
            if(diamondBoard[row-1][column]=='*')
                diaCounter++;
            userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(column==7 && row>=2 && row<=6)
    {
        if(diamondBoard[row-2][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row][column-1]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }
    else if(row>=2 && row<=6 && column>=2 && column<=6)
    {
        if(diamondBoard[row-2][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row-2][column]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row-1][column]=='*')
```

```
            diaCounter++;
        if(diamondBoard[row][column-2]=='*')
            diaCounter++;
        if(diamondBoard[row][column-1]=='*')
            diaCounter++;
        if(diamondBoard[row][column]=='*')
            diaCounter++;
        userBoard[row-1][column-1]=diaCounter+48;
    }

}

void displayUserBoard(int **board)
{
    printf("\n         UserBoard\n\n");
    int i,j;
    printf("     1  2  3  4  5  6  7\n");
    for(i=0;i<ROW;i++)
    {
        printf(" \n %d  ",i+1);
        for(j=0;j<COLUMN;j++)
        {



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

        }

    }

    printf("\n");

}

void displayDiamondBoard(int **board)
{
    printf("\n      DiamondBoard\n");
    int i,j;

    for(i=0;i<ROW;i++)
    {
        printf("\n");
        for(j=0;j<COLUMN;j++)
            printf(" %c ",board[i][j]);
    }
    printf("\n");
}
```