

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<time.h>
4
5
6  #define ROW 7
7  #define COLUMN 7
8  #define ROUNDS 10
9  #define TRUE 1
10 #define FALSE 0
11
12 // 42 = *
13 // 63 = ?
14
15 void initializeBoards(int**,int**,int);
16 int randomDiamond();
17 int getGuessedCoordinates(int,int);
18 int checkDiamonds(int,int,int**);
19 void provideHints(int,int,int**,int**);
20 void displayUserBoard(int**);
21 void displayDiamondBoard(int **);
22
23 int main()
24 {
25
26 srand(time(NULL));
27 int **userBoard, **diamondBoard;
28 int i,n;//number of diamonds
29 int rowCoordinate,columnCoordinate;
30 int totalPoint=0,bestScore=0;
31 char playAgain;
32
33 userBoard=(int**)malloc(ROW*sizeof(int*));
34 diamondBoard=(int**)malloc(ROW*sizeof(int*));
35
36 if(userBoard==NULL || diamondBoard==NULL)
37 {
38     printf("Out of memory!");
39     return 0;
40 }
41 for(i=0;i<ROW;i++)
42 {
43     userBoard[i]=(int*)malloc(COLUMN*sizeof(int));
44     diamondBoard[i]=(int*)malloc(COLUMN*sizeof(int));
45     if(userBoard[i]==NULL || diamondBoard[i]==NULL)
46     {
47         printf("Out of memory!");
48         return 0;
49     }
50 }
51
52 int roundCounter=1;
53 int diamondCounter=0;
54
55
56 while(roundCounter<=ROUNDS)
57 {
58     if(roundCounter==1)
59     {
60         printf("\n*Diamond-Hunter*\nLets get started!\n");
61         printf("Enter the number of diamonds to hunt:");
62         scanf("%d",&n);
63         while(n>49 || n<1)
64         {
65             printf("please enter number of diamonds between 1-49:");
66             scanf("%d",&n);

```

```

67     }
68     initializeBoards(userBoard,diamondBoard,n);
69     displayUserBoard(userBoard);
70 }
71
72 printf("\n\n");
73 printf("Round %d:\n",roundCounter);
74 printf("Enter the coordinates of the diamonds: ");
75 scanf("%d %d",&rowCoordinate,&columnCoordinate);
76 fflush(stdin);
77
78 int checkGuessedCoordinates=getGuessedCoordinates(rowCoordinate,columnCoordinate);
79 while(checkGuessedCoordinates==0)
80 {
81     printf("Please enter coordinates between 1-7! ");
82     scanf("%d %d",&rowCoordinate,&columnCoordinate);
83     fflush(stdin);
84     checkGuessedCoordinates=getGuessedCoordinates(rowCoordinate,columnCoordinate);
85 }
86
87 int checkDia=checkDiamonds(rowCoordinate,columnCoordinate,diamondBoard);
88 if(checkDia==1)
89 {
90     printf("You got 100 points!\n");
91     totalPoint+=100;
92     diamondCounter++;
93 }
94 else
95 {
96     printf("You got 0 points!\n");
97 }
98
99 provideHints(rowCoordinate,columnCoordinate,userBoard,diamondBoard);
100 printf("Your total points is %d!\n",totalPoint);
101 displayUserBoard(userBoard);
102 if(diamondCounter==n || roundCounter==ROUNDS)
103 {
104     if(totalPoint>bestScore)
105         bestScore=totalPoint;
106     if(diamondCounter==n)
107         printf("\nYou won! You found all diamonds!\n");
108     else
109         printf("\nYou run out of rounds! Game over!\n");
110
111     printf("Diamond Board is as follows:\n");
112     displayDiamondBoard(diamondBoard);
113     printf("\nYour score is %d\n",totalPoint);
114     printf("Your best score is %d\n",bestScore);
115     printf("Do you want to play again?(Y/N)");
116     scanf("%c",&playAgain);
117     fflush(stdin);
118     while(playAgain!='Y' && playAgain!='N')
119     {
120         printf("Please enter 'Y' for YES, 'N' for NO!\n");
121         printf("Do you want to play again?(Y/N)");
122         scanf("%c",&playAgain);
123         fflush(stdin);
124     }
125     if(playAgain=='Y')
126     {
127         roundCounter=0;
128         totalPoint=0;
129         diamondCounter=0;
130     }
131     else
132     {

```

```

133         printf("\nSee you!\n");
134         break;
135     }
136
137 }
138 roundCounter++;
139 }
140
141 return 0;
142 }
143
144 void initializeBoards(int** userBoard,int** diamondBoard,int n)
145 {
146     int i,j,k;
147     int coordinate1,coordinate2;
148
149     for(i=0;i<ROW;i++)
150     {
151         for(j=0;j<COLUMN;j++)
152         {
153             userBoard[i][j]=63;
154         }
155     }
156     for(i=0;i<ROW;i++)
157     {
158         for(j=0;j<COLUMN;j++)
159         {
160             diamondBoard[i][j]=63;
161         }
162     }
163
164     for(k=0;k<n;k++)
165     {
166         coordinate1=randomDiamond();
167         coordinate2=randomDiamond();
168
169         for(i=0;i<ROW;i++)
170         {
171             for(j=0;j<COLUMN;j++)
172             {
173                 if(i==coordinate1 && j==coordinate2 && diamondBoard[i][j]!='*')
174                     diamondBoard[i][j]=42;
175                 else if(i==coordinate1 && j==coordinate2 && diamondBoard[i][j]=='*')
176                 {
177                     k--;
178                 }
179             }
180         }
181     }
182 }
183
184
185
186
187
188 }
189
190 int randomDiamond()
191 {
192     return rand()%7;
193 }
194
195 int getGuessedCoordinates(int row,int column)
196 {
197     if(row<1 || row>7 || column<1 || column>7)
198         return FALSE;

```

```

199     else
200         return TRUE;
201 }
202
203 int checkDiamonds(int row, int column, int **diamondBoard)
204 {
205
206     if(diamondBoard[row-1][column-1]=='*')
207         return TRUE;
208     else
209         return FALSE;
210 }
211
212 void provideHints(int row, int column, int** userBoard, int** diamondBoard)
213 {
214
215     int i,j;
216     int diaCounter=0;
217
218     if(diamondBoard[row-1][column-1]=='*')
219     {
220         userBoard[row-1][column-1]='*';
221     }
222     else if(row==1 && column==1)
223     {
224         if(diamondBoard[row][column-1]=='*')
225             diaCounter++;
226         if(diamondBoard[row][column]=='*')
227             diaCounter++;
228         if(diamondBoard[row-1][column]=='*')
229             diaCounter++;
230         userBoard[row-1][column-1]=diaCounter+48;
231     }
232
233     else if(row==7 && column==1)
234     {
235         if(diamondBoard[row-2][column-1]=='*')
236             diaCounter++;
237         if(diamondBoard[row-2][column]=='*')
238             diaCounter++;
239         if(diamondBoard[row-1][column]=='*')
240             diaCounter++;
241         userBoard[row-1][column-1]=diaCounter+48;
242     }
243     else if(row==1 && column==7)
244     {
245         if(diamondBoard[row-1][column-2]=='*')
246             diaCounter++;
247         if(diamondBoard[row][column-2]=='*')
248             diaCounter++;
249         if(diamondBoard[row][column-1]=='*')
250             diaCounter++;
251         userBoard[row-1][column-1]=diaCounter+48;
252     }
253     else if(row==7 && column==7)
254     {
255         if(diamondBoard[row-2][column-2]=='*')
256             diaCounter++;
257         if(diamondBoard[row-2][column-1]=='*')
258             diaCounter++;
259         if(diamondBoard[row-1][column-2]=='*')
260             diaCounter++;
261         userBoard[row-1][column-1]=diaCounter+48;
262     }
263
264     else if(row==1 && column>=2 && column<=6)

```

```

265 {
266     if(diamondBoard[row-1][column-2]=='*')
267         diaCounter++;
268     if(diamondBoard[row][column-2]=='*')
269         diaCounter++;
270     if(diamondBoard[row][column-1]=='*')
271         diaCounter++;
272     if(diamondBoard[row][column]=='*')
273         diaCounter++;
274     if(diamondBoard[row-1][column]=='*')
275         diaCounter++;
276     userBoard[row-1][column-1]=diaCounter+48;
277 }
278 else if(column==1 && row>=2 && row<=6)
279 {
280     if(diamondBoard[row-2][column-1]=='*')
281         diaCounter++;
282     if(diamondBoard[row-2][column]=='*')
283         diaCounter++;
284     if(diamondBoard[row-1][column]=='*')
285         diaCounter++;
286     if(diamondBoard[row][column]=='*')
287         diaCounter++;
288     if(diamondBoard[row][column-1]=='*')
289         diaCounter++;
290     userBoard[row-1][column-1]=diaCounter+48;
291 }
292 else if(row==7 && column>=2 && column<=6)
293 {
294     if(diamondBoard[row-1][column-2]=='*')
295         diaCounter++;
296     if(diamondBoard[row-2][column-2]=='*')
297         diaCounter++;
298     if(diamondBoard[row-2][column-1]=='*')
299         diaCounter++;
300     if(diamondBoard[row-2][column]=='*')
301         diaCounter++;
302     if(diamondBoard[row-1][column]=='*')
303         diaCounter++;
304     userBoard[row-1][column-1]=diaCounter+48;
305 }
306 else if(column==7 && row>=2 && row<=6)
307 {
308     if(diamondBoard[row-2][column-1]=='*')
309         diaCounter++;
310     if(diamondBoard[row-2][column-2]=='*')
311         diaCounter++;
312     if(diamondBoard[row-1][column-2]=='*')
313         diaCounter++;
314     if(diamondBoard[row][column-2]=='*')
315         diaCounter++;
316     if(diamondBoard[row][column-1]=='*')
317         diaCounter++;
318     userBoard[row-1][column-1]=diaCounter+48;
319 }
320 else if(row>=2 && row<=6 && column>=2 && column<=6)
321 {
322     if(diamondBoard[row-2][column-2]=='*')
323         diaCounter++;
324     if(diamondBoard[row-2][column-1]=='*')
325         diaCounter++;
326     if(diamondBoard[row-2][column]=='*')
327         diaCounter++;
328     if(diamondBoard[row-1][column-2]=='*')
329         diaCounter++;
330     if(diamondBoard[row-1][column]=='*')

```

```

331         diaCounter++;
332         if(diamondBoard[row][column-2]=='*')
333             diaCounter++;
334         if(diamondBoard[row][column-1]=='*')
335             diaCounter++;
336         if(diamondBoard[row][column]=='*')
337             diaCounter++;
338         userBoard[row-1][column-1]=diaCounter+48;
339     }
340
341 }
342
343 void displayUserBoard(int **board)
344 {
345     printf("\n          UserBoard\n\n");
346     int i,j;
347     printf("          1  2  3  4  5  6  7\n");
348     for(i=0;i<ROW;i++)
349     {
350         printf(" \n %d  ",i+1);
351         for(j=0;j<COLUMN;j++)
352         {
353
354
355
356             printf(" %c ",board[i][j]);
357
358         }
359     }
360
361     printf("\n");
362 }
363
364 void displayDiamondBoard(int **board)
365 {
366     printf("\n          DiamondBoard\n");
367     int i,j;
368
369     for(i=0;i<ROW;i++)
370     {
371         printf("\n");
372         for(j=0;j<COLUMN;j++)
373             printf(" %c ",board[i][j]);
374     }
375     printf("\n");
376 }
377
378
379
380
381
382
383
384
385

```