

```

1  #include <stdio.h>
2  #define X 8
3  #define Y 8
4  int road[X][Y]; //存储某个点是第几步
5  int nextxy(int *x,int *y,int count)
6  {
7      switch(count)
8      {
9          case 0:
10             if(*x+2<X && *y-1>=0 && road[*x+2][*y-1]==0)
11             {
12                 *x+=2;
13                 *y-=1;
14                 return 1;
15             }
16             break;
17          case 1:
18             if(*x+2<X && *y+1<Y && road[*x+2][*y+1]==0)
19             {
20                 *x+=2;
21                 *y+=1;
22                 return 1;
23             }
24             break;
25          case 2:
26             if(*x+1<X && *y-2>=0 && road[*x+1][*y-2]==0)
27             {
28                 *x+=1;
29                 *y-=2;
30                 return 1;
31             }
32             break;
33          case 3:
34             if(*x+1<X && *y+2<Y && road[*x+1][*y+2]==0)
35             {
36                 *x+=1;
37                 *y+=2;
38                 return 1;
39             }
40             break;
41          case 4:
42             if(*x-2>=0 && *y-1>=0 && road[*x-2][*y-1]==0)
43             {
44                 *x-=2;
45                 *y-=1;
46                 return 1;
47             }
48             break;
49          case 5:
50             if(*x-2>=0 && *y+1<Y && road[*x-2][*y+1]==0)
51             {
52                 *x-=2;
53                 *y+=1;
54                 return 1;
55             }

```

```

56         break;
57     case 6:
58         if(*x-1>=0 && *y-2>=0 && road[*x-1][*y-2]==0)
59         {
60             *x-=1;
61             *y-=2;
62             return 1;
63         }
64         break;
65     case 7:
66         if(*x-1>=0 && *y+2<Y && road[*x-1][*y+2]==0)
67         {
68             *x-=1;
69             *y+=2;
70             return 1;
71         }
72         break;
73     default:
74         break;
75     }
76     return 0;
77 }
78
79 //输出计算出来的走法，输出的矩阵元素值表示走的第road[i][j]步
80 void print()
81 {
82     int i,j;
83     for(i=0;i<X;++i)
84     {
85         for(j=0;j<Y;++j)
86         {
87             printf("%2d\t",road[i][j]);
88         }
89         printf("\n");
90     }
91     printf("\n");
92 }
93
94 //tag为标记变量,每走一步, tag+1
95 int TravelroadBoard(int x,int y,int tag)
96 {
97     int x1=x,y1=y,flag=0;
98     int count=0; // 对现在的位置，每次从位置0出开始走
99     road[x][y]=tag;
100     if(tag==x*Y)
101     {
102         print();//打印棋盘
103         return 1;
104     }
105     //确定成功走出第一步
106     flag=nextxy(&x1,&y1,count);
107     while(flag==0 && count<=7)
108     {
109         ++count; // 若行不通，则切换走的方向
110         flag=nextxy(&x1,&y1,count);
111     }
112     while(flag)
113     {

```

```

114         if(TravelroadBoard(x1,y1,tag+1)) // 成功走完全部
115             return 1;
116         // 未成功走完, 换个方向重新开始走
117         x1=x;y1=y;
118         ++count;
119         flag=nextxy(&x1,&y1,count);
120         while(flag==0 && count<7)
121         {
122             ++count;
123             flag=nextxy(&x1,&y1,count);
124         }
125     }
126     if(flag==0)
127     {
128         road[x][y]=0;
129     }
130     return 0;
131 }
132
133 int main()
134 {
135     int i,j;
136     for(i=0;i<X;++i)
137     {
138         for(j=0;j<Y;++j)
139             road[i][j]=0;
140     }
141     if(!TravelroadBoard(2,0,1)) //起始点(2,0)
142     {
143         printf("抱歉, 马踏棋盘失败鸟~\n");
144     }
145     return 0;
146 }

```

```

PS D:\csjjg> cd "d:\csjjg\程序设计综合实践\第四次作业\" ; if ($?) { g
cc 马踏棋盘.c -o 马踏棋盘 } ; if ($?) { .\马踏棋盘 }

```

43	50	47	38	57	52	61	32
48	37	44	51	46	33	58	53
1	42	49	56	39	60	31	62
36	15	40	45	34	29	54	59
41	2	35	16	55	24	63	30
14	5	12	9	22	19	28	25
3	10	7	20	17	26	23	64
6	13	4	11	8	21	18	27