

In [4]:

```
1 import random
2 # n=8
3 #n-queen using hill-climbing method
4 N=8
5
6 def printBoard(board):
7     for i in range(N):
8         for j in range(N):
9             print(board[i][j], end=" ")
10            print("\n")
11        print("\n")
12
13
14 def fill(board):
15     for i in range(N):
16         for j in range(N):
17             board[i][j] = 0
18
19
20 def copyState(state1, state2):
21     for i in range(N):
22         state1[i] = state2[i]
23
24
25 def generateBoard(board, State):
26     fill(board)
27     for i in range(N):
28         board[state[i]][i] = 1
29
30 def compareState(state1, state2):
31     for i in range(N):
32         if(state1[i] != state2[i]):
33             return False
34     return True
35
36 def calculateObjective(board, state):
37     cnt=0
38     row=0
39     col=0
40     for i in range(N):
41         row= state[i]
42         col= i-1
43         while(col>=0):
44             if(board[row][col] == 1):
45                 cnt+=1
46             col-=1
47
48         row= state[i]
49         col= i+1
50         while(col<N):
51             if(board[row][col] == 1):
52                 cnt+=1
53             col+=1
54
55         row= state[i]-1
56         col = i-1
57         while (col>= 0 and row>=0):
58             if(board[row][col] == 1):
59                 cnt+=1
```

```

60         col-=1
61         row-=1
62
63     row = state[i]+1
64     col= i+1
65     while (col < N and row < N):
66         if(board[row][col] == 1):
67             cnt+=1
68             col+=1
69             row+=1
70
71     row = state[i]+1
72     col= i-1
73     while (col>=0 and row < N):
74         if(board[row][col] == 1):
75             cnt+=1
76             col-=1
77             row+=1
78
79     row = state[i]-1
80     col= i+1
81     while (col < N and row>=0):
82         if(board[row][col] == 1):
83             cnt+=1
84             col+=1
85             row-=1
86
87
88     return cnt/2
89
90
91 def getNeighbour(board, state):
92     opState= [0 for i in range(N)]
93     opBoard=[[0 for i in range(N)] for j in range(N)]
94     copyState(opState, state)
95     generateBoard(opBoard, opState)
96
97     opObjective= calculateObjective(opBoard, opState)
98
99     neighbourState= [0 for i in range(N)]
100    neighbourBoard=[[0 for i in range(N)] for j in range(N)]
101    copyState(neighbourState, state)
102    generateBoard(neighbourBoard, neighbourState)
103
104    for i in range(N):
105        for j in range(N):
106            if(j!= state[i]):
107                neighbourState[i] = j
108                neighbourBoard[neighbourState[i]][i]= 1
109                neighbourBoard[state[i]][i]= 0
110
111                temp= calculateObjective(neighbourBoard, neighbourState)
112
113                if(temp< opObjective):
114                    opObjective= temp
115                    copyState(opState, neighbourState)
116                    generateBoard(opBoard, opState)
117
118
119    neighbourBoard[neighbourState[i]][i]= 0
120    neighbourState[i] = state[i]

```

```

121         neighbourBoard[state[i]][i]= 1
122
123     copyState(state, opState)
124     generateBoard(board, state)
125
126
127 def hillCimbing(board, state):
128     neighbourState= [0 for i in range(N)]
129     neighbourBoard=[[0 for i in range(N)] for j in range(N)]
130     copyState(neighbourState, state)
131     generateBoard(neighbourBoard, neighbourState)
132
133     while(True):
134         copyState(state, neighbourState)
135         generateBoard(board, state)
136         getNeighbour(neighbourBoard, neighbourState)
137         if(compareState(state, neighbourState)):
138             printBoard(board)
139             break
140         if(calculateObjective(board, state)==calculateObjective(neighbourBoard
141             neighbourState[random.randrange(0, N)]= random.randrange(0, 8)
142             generateBoard(neighbourBoard, neighbourState)
143
144
145
146
147 state= [0 for i in range(N)]
148 board=[[0 for i in range(N)] for j in range(N)]
149
150 #configure randomly
151 for i in range(0, N):
152     state[i]= random.randrange(0, 8)
153     board[state[i]][i]= 1
154
155 hillCimbing(board, state)
156

```

```

1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 1
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0

```