

CSP: N - Queens Problem

```
n_queen.py > ...
1
2  global N
3  N = 4
4
5  def printSolution(board):
6      for i in range(N):
7          for j in range(N):
8              if board[i][j] == 1:
9                  print("Q",end=" ")
10             else:
11                 print(".",end=" ")
12             print()
13
14
15  def isSafe(board, row, col):
16
17      for i in range(col):
18          if board[row][i] == 1:
19              return False
20
21      for i, j in zip(range(row, -1, -1),
22                    range(col, -1, -1)):
23          if board[i][j] == 1:
24              return False
25
26      for i, j in zip(range(row, N, 1),
27                    range(col, -1, -1)):
28          if board[i][j] == 1:
29              return False
30
31      return True
32
33
34  def solveNQutil(board, col):
35
36      if col >= N:
37          return True
```

```

39     for i in range(N):
40
41         if isSafe(board, i, col):
42
43             board[i][col] = 1
44
45             if solveNQutil(board, col + 1) == True:
46                 return True
47
48             board[i][col] = 0
49
50     return False
51
52
53 def solveNQ():
54     board = [[0, 0, 0, 0],
55             [0, 0, 0, 0],
56             [0, 0, 0, 0],
57             [0, 0, 0, 0]]
58
59     if solveNQutil(board, 0) == False:
60         print("Solution does not exist")
61         return False
62
63     printSolution(board)
64     return True
65
66
67 # Driver Code
68 if __name__ == '__main__':
69     solveNQ()
70
71
72

```

Output:

```

PS C:\Users\abhib\Desktop\AI> & C:/Users/abhib/Desktop/ml_project/env/python.exe c:/Users/abhib/Desktop/AI/n_queen.py
. . Q .
Q . . .
. . . Q
. Q . .
PS C:\Users\abhib\Desktop\AI>

```