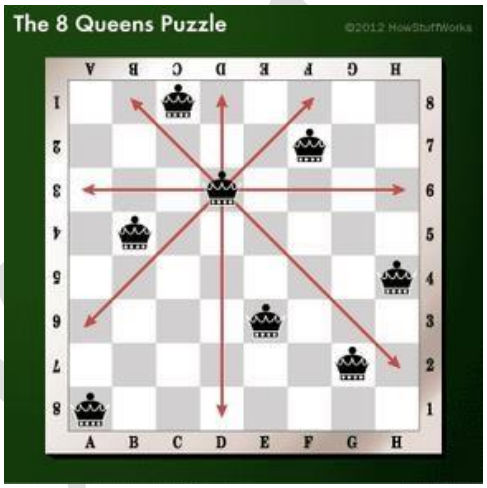
EX.NO: 1

8- QUEENS PROBLEM

You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other queen on the chessboard. A queen can only be attacked if it lies on the same row, or same column, or the same diagonal of any other queen. Print all the possible configurations.

To solve this problem, we will make use of the Backtracking algorithm. The backtracking algorithm, in general checks all possible configurations and test whether the required result is obtained or not. For the given problem, we will explore all possible positions the queens can be relatively placed at. The solution will be correct when the number of placed queens = 8.



AIM :

To implement an 8-Queesns problem using python.

SOURCE CODE :

```
def isSafe(mat, r, c):      for i in
range(r):         if mat[i][c] == 'Q':
return False       (i, j) = (r, c)
while i >= 0 and j >= 0:            if
mat[i][j] == 'Q':                 return
False           i = i - 1            j = j -
1       (i, j) = (r, c)          while i >= 0
and j < len(mat):               if
mat[i][j] == 'Q':                    return
False              i = i - 1
            j    =    j    +    1
return      True          def
printSolution(mat):
for r in mat:
       print(str(r).replace(',', '').replace('\'', ''))
print()  def nQueen(mat, r):        if r ==
len(mat):            printSolution(mat)
return       for i in range(len(mat)):            if
isSafe(mat, r, i):                mat[r][i] = 'Q'
nQueen(mat, r + 1)                  mat[r][i] = '–'
if __name__ == '__main__':
 N = int(input("Enter no of Queens you want : ")) mat
= [['–' for x in range(N)] for y in range(N)]
nQueen(mat, 0)
```

OUTPUT:

```
Enter no of Queens you want : 8
[Q - - - - - - -]
[- - - - Q - - -]
[- - - - - - - Q]
[- - - - - Q - -]
[- - Q - - - - -]
[- - - - - - Q -]
[- Q - - - - - -]
[- - - Q - - - -]
```


RESULT: Thus the above python code is executed successfully and output is verified.