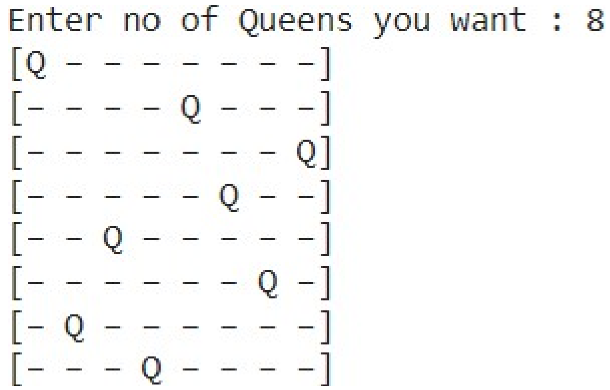
AIM :

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 implementation an 8-Queesns problem using python and the outputs have been verified .