N-Queens Problem

for y m mande (N)

del 1550/c (board, now, col):

in mange (col):

if board [1]

/lim:

To solve the N-Queens problem by identifying all valid configurations of originating atleast one solution, depending on the originarement.

Algorithm:

\$ 1. Start

- 2. Create an NEN board with all positions set to O (empty)
- 3. For each column, try placing a grucen in
- position (is safe using the isSafe()
- 5. If placing the queen is safe, mark the position with al
- 6. Recursively move to the reset column and repeat
- 7. If no valid placement is found, backtrack by removing the queen, and by a new position. position.
- 8. If queens are placed in all columns, print
- possible point net-win False

10-Stop

```
def posintsolution (board): ) -
                                         for i in range(N):
                                                     for j in range(N):
                                                                   if board [i] [j] == 1:
                printitude pod else: "I" end = ") als subse of
                           desertion print (of of, endetway) in the solution of printegenest.
                          def is Safe (board, now, col):
                                                                                                                                                                                          Mornthum
                                    for i in range (col):
                                                     if board[row ][i] == 1:
                                                                     return False
                                                                                                                                                  1 -1) nange (col, -1,-1);
                                     Jon i din zipl nangel now,
                                                       if board [i] EjJ Fitch ) O Jost toz
            neturn False mulas dans root. E

no nous proposition p
Li Check if the presidential bound by is Safe()
                                                                 neturn False
                               5. If placing the queen is sufet to most en
                                                                                                        position with al
                  def Solve Na Util (board, is col):
                                                    return True
         the topological biles on the transce (N):

In range (N):

In serious primarie policy is is Safe (board, i, col):

The primarie policy is is safe (board, i, col):

The primarie policy is in the primarie policy in the primarie policy is safe (board, i, col):
            board [i][col]=1

tring some of Solve NQ Util (board, col+1):
                                                                           return True
                                  rel-un False
                                              rel-un False
                                                                                                                                                            10-Stop
```

del SelveNQ(): lepth First global N N = **(input()) board [[0 for i in range(N)] for j in range(N)) if Solve NQUEIL (board, 0) == False: point ("Solution does not exist") return False Megaritim point Solution (board) return True SolveNQ() o set to keep track of starting node Output: as visited Enter the number : 3 Solution does not exist 270 Enter the number

Result:

The program has been executed successfully and the output has been verified.