ARTIFICIAL INTELLIGENCE
LAB 1 - TOY PROBLEM

RA1811003010516
Tanmay Toshniwal

Problem: The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other. A queen can attack other queens either diagonally or in same row and column. Generate all possible configurations of queens on board and print configurations that satisfy the given constraints.

Solution (Python):

```
def initialize(n):
    for key in ['queen', 'row', 'col', 'nwtose', 'swtone']:
        board[key] = {}
    for i in range(n):
        board['queen'][i] = -1
        board['row'][i] = 0
        board['col'][i] = 0
    for i in range(-(n - 1), n):
        board['nwtose'][i] = 0
    for i in range(2 * n - 1):
        board['swtone'][i] = 0


def printboard():
    for row in sorted(board['queen'].keys()):
        print((row, board['queen'][row]), end=" ")
    print("")


def free(i, j):
    return (board['row'][i] == 0 and board['col'][j] == 0 and
            board['nwtose'][j - i] == 0 and board['swtone'][j + i] == 0)


def addqueen(i, j):
    board['queen'][i] = j
    board['row'][i] = 1
    board['col'][j] = 1
    board['nwtose'][j - i] = 1
    board['swtone'][j + i] = 1
```

```
def undoqueen(i, j):
    board['queen'][i] = -1
    board['row'][i] = 0
    board['col'][j] = 0
    board['nwtose'][j - i] = 0
    board['swtone'][j + i] = 0


def placequeen(i):
    n = len(board['queen'].keys())
    for j in range(n):
        if free(i, j):
            addqueen(i, j)
            if i == n - 1:
                printboard()
            else:
                extendsoln = placequeen(i + 1)
            undoqueen(i, j)


board = {}
n = int(input("How many queens? "))
initialize(n)
if placequeen(0):
    printboard()
```

Input: `4`

Output: `(0, 1) (1, 3) (2, 0) (3, 2)`
`(0, 2) (1, 0) (2, 3) (3, 1)`

- What is a toy problem?

  It is a concise and exact description of the problem which is used by the researchers to compare the performance of algorithms.

- How many approaches do you have for solving the toy problem which you have taken?

  The Eight Queens problem can be solved using recursion, simple brute force search, backtracking depth first search, heuristics and also genetic algorithms.

- Map your toy problem to any real life application.

  The N-queen problem is used in many practical solutions like parallel memory storage schemes, VLSI testing, traffic control and deadlock prevention. This problem is also used to find out solutions to more practical problems which requires permutation like travelling salesman problem.