Experiment - 1

N - Queens

AIM

To solve the N-Queens problem using the backtracking algorithm in Python, where the goal is to place N queens on a N×N chess board such that no two queens threaten each other. The program will take N as input.

Algorithm

1. Create an N×N board initialized with zeroes(0). Each cell represents an empty spot where a queen might be placed.
2. Create a recursive function solveNQueens(board, col) that attempts to place queens on the board column by column.
3. If the column index equals N, it equals N, it means all queens are successfully placed, and we have found a solution.
4. Create a helper function isSafe(board, row, col) to check whether placing a queen at board[row][col] is safe. Ensure no other queens are on the same row, column, or diagonal.
5. If placing a queen in any row of the current column doesn't lead to a solution, backtrack by removing the queen and trying the next row.
6. Once all queens are placed successfully, add the board configuration to the list of solutions.

7. Continue searching for solutions by backtracking to find all possible ways to place N queens.
8. Take input N from the user to determine the size of the board and the number of queens.

Program

```
def isSafe(board, row, col, N):
    for i in range(col):
        if board[row][i]==1:
            return False

    for i,j in zip(range(row,-1,-1), range(col,-1,-1)):
        if board[i][j]==1:
            return False

    for i,j in zip(range(row,N), range(col,-1,-1)):
        if board[i][j]==1:
            return False
    return True

def solveQueens(board, col, N):
    if col>=N:
        return True
    for i in range(N):
        if isSafe(board,i,col,N):
            board[i][col]=1
            if solveNQueens(board,col+1,N):
                return True
            board[i][col]=0
    return False
```

def

```
def printSolution(board,N):
    for i in range(N):
        for j in range(N):
            if board[i][j]==1:
                print("Q",end=" ")
            else:
                print(".",end=" ")
        print()
    print("\n")

def solveNQueensProblem(N):
    board=[[0 for _ in range(N)] for _ in range(N)]
    if not solveNQueens(board,0,N):
        print("Solution does not exist")
        return False
    printSolution(board,N)
    return True

if __name__=="__main__":
    N=int(input("Enter N:"))
    solveNQueensProblem(N)
```

Output

```
Enter N: 4
. . Q .
Q . . .
. . . Q
. Q . .
```

Result

The N-Queens problem was successfully solved using the backtracking algorithm. The output is verified.