Name : L.vijay antony

Roll no : 241801311

Date : 28/03/2025

Ex no : 01

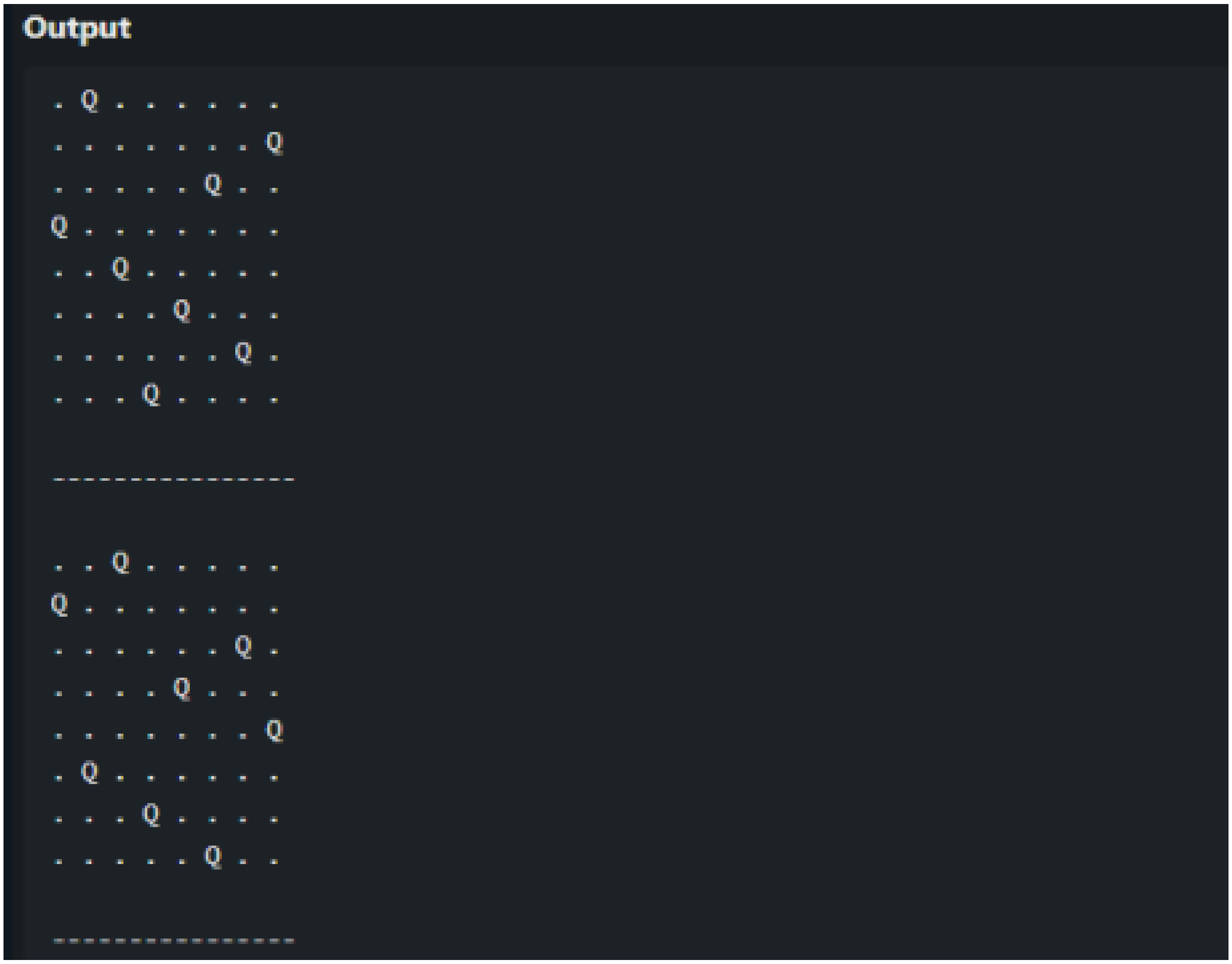
Ex name : 8 Queens problem

Program:

```
N = 8 def print_board(board): for row in board: print(" ".join(row)) print("\n" + "-" * 16 + "\n")

def is_safe(board, row, col):

for i in range(row):
 if board[i][col] == "Q":
 return False

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

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

 return True


def solve(board, row): if row == N: print_board(board) return True

 res = False
for col in range(N):
 if is_safe(board, row, col):
 board[row][col] = "Q"
 res = solve(board, row + 1) or res
 board[row][col] = "."
return res
```

```
def solve_n_queens(): board = [["." for _ in range(N)] for _ in range(N)] if not
solve(board, 0): print("No solution exists")

solve_n_queens()
```

Output:



```
. Q . . . . . .
. . . . . . . Q
. . . . . Q . .
Q . . . . . . .
. . Q . . . . .
. . . . Q . . .
. . . . . . Q .
. . . Q . . . .

----------------

. . Q . . . . .
Q . . . . . . .
. . . . . . Q .
. . . . Q . . .
. . . . . . . Q
. Q . . . . . .
. . . Q . . . .
. . . . . Q . .

----------------
```