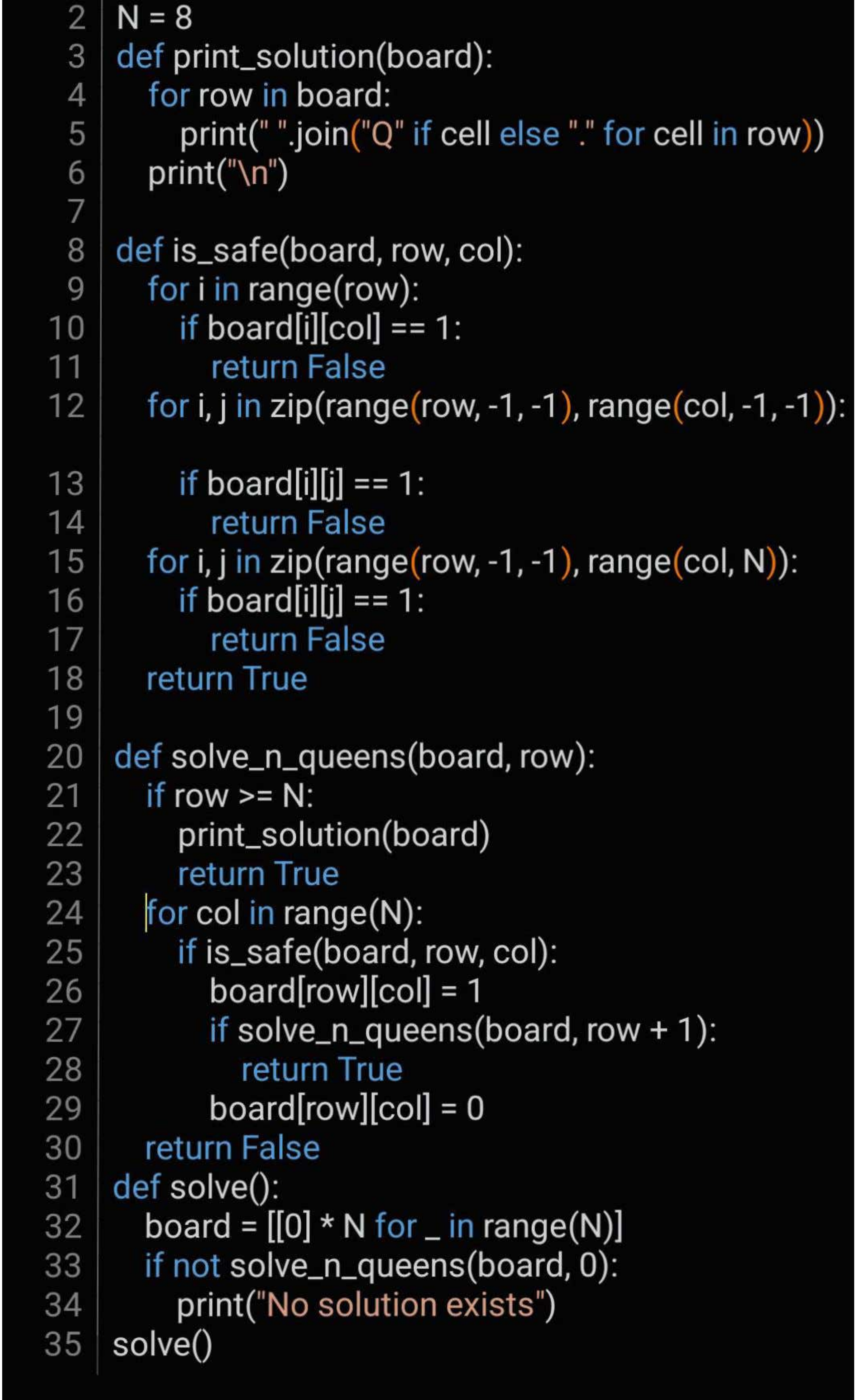
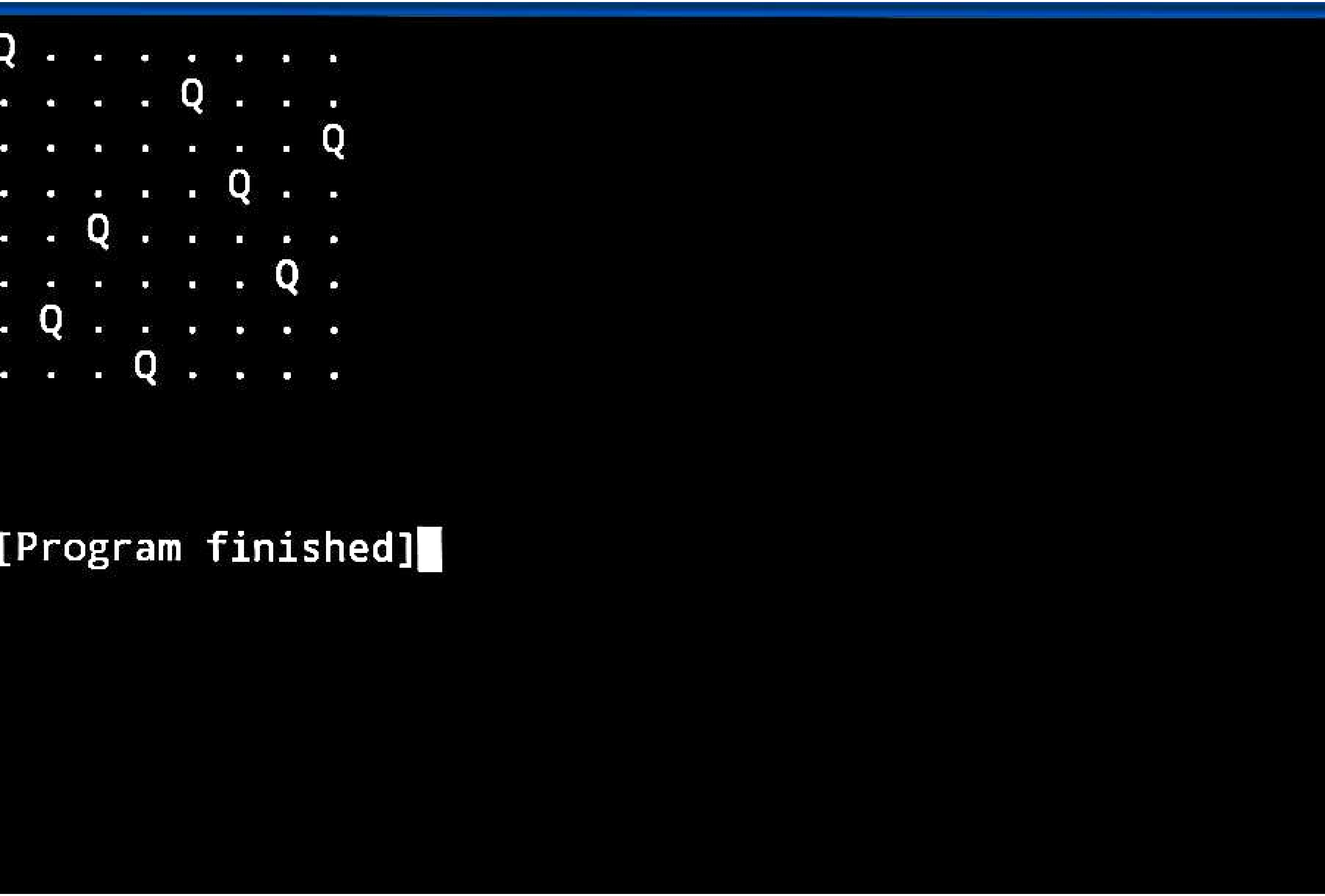


```python
N = 8
def print_solution(board):
    for row in board:
        print(" ".join("Q" if cell else "." for cell in row))
    print("\n")

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

def solve_n_queens(board, row):
    if row >= N:
        print_solution(board)
        return True
    for col in range(N):
        if is_safe(board, row, col):
            board[row][col] = 1
            if solve_n_queens(board, row + 1):
                return True
            board[row][col] = 0
    return False
def solve():
    board = [[0] * N for _ in range(N)]
    if not solve_n_queens(board, 0):
        print("No solution exists")
solve()
```



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



[Program finished]
```