Assignment NO – 5 (DAA) (LP-III)

**Code**

def is\_safe(board, row, col, n):

# Check if there is a queen in the same column

for i in range(row):

if board[i][col] == 1:

return False

# Check upper left diagonal

for i, j in zip(range(row, -1, -1), range(col, -1, -1)):

if board[i][j] == 1:

return False

# Check upper right diagonal

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, n):

if row == n:

return True

for col in range(n):

if is\_safe(board, row, col, n):

board[row][col] = 1

if solve\_n\_queens(board, row + 1, n):

return True

board[row][col] = 0

return False

def print\_board(board):

for row in board:

print(' '.join(['Q' if cell == 1 else '.' for cell in row]))

def main():

n = int(input("Enter the value of N for N-Queens: "))

board = [[0 for \_ in range(n)] for \_ in range(n)]

if solve\_n\_queens(board, 0, n):

print("N-Queens solution:")

print\_board(board)

else:

print(f"No solution exists for {n}-Queens.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Output**

Enter the value of N for N-Queens: 4

N-Queens solution:

Q . . .

. . Q .

. Q . .

. . . Q