Question:2

8-QUEENS PROBLEM

AIM

To solve the 8-Queens problem using python

ALGORITHM

1. Define a function `is_safe(board, row, col)` to check if it's safe to place a queen at a given position `(row, col)` on the board.
   - Check if there is no queen in the same column.
   - Check if there is no queen in the upper left diagonal.
   - Check if there is no queen in the upper right diagonal.
2. Define a function `solve_queens(board, row)` to recursively place queens on the board.
   - Base case: If all queens are placed (row equals the length of the board), return `True`.
   - Iterate through each column.
   - If it's safe to place a queen at `(row, col)`, mark it as placed (1) on the board and recursively solve for the next row.
   - If placing a queen in this column leads to a solution, return `True`.
   - If not, backtrack by resetting the board at `(row, col)` to 0 and try the next column.
3. Define a function `print_solution(board)` to print the final board configuration.
4. Define a `main()` function to initialize an empty board, call the `solve_queens()` function, and print the solution if found.
5. In the main block, call the `main()` function if the script is run as the main program.

CODE

```
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, len(board))):
        if board[i][j] == 1:
            return False
    return True
```

```
def solve_queens(board, row):
    if row == len(board):
        return True

    for col in range(len(board)):
        if is_safe(board, row, col):
            board[row][col] = 1
            if solve_queens(board, row + 1):
                return True
            board[row][col] = 0
    return False

def print_solution(board):
    for row in board:
        print(" ".join(map(str, row)))

def main():
    board = [[0] * 8 for _ in range(8)]
    if solve_queens(board, 0):
        print("Solution found:")
        print_solution(board)
    else:
        print("No solution exists.")

if __name__ == "__main__":
    main()
```

OUTPUT

```
1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0
```