Write the python program to solve 8-Queen problem

AIM

To implement a Python program that solves the **8-Queens Problem** using **Backtracking** technique.

ALGORITHM

1. Start with an empty 8×8 chessboard.
2. Place a queen in a row one by one.
3. Before placing a queen in a column, check if it is **safe** (no other queen in the same column, and both diagonals).
4. If safe, place the queen and move to the next row.
5. If no safe position exists in the current row, backtrack to the previous row and move the queen to the next column.
6. Repeat until all 8 queens are placed on the board.
7. Print all valid solutions.

8 QUEEN AI PYTHON.py - C:/Users/gayathri/Downloads/8 QUEEN AI PYTHON.py (3.8.2)

File Edit Format Run Options Window Help

```
def print_board(board):
    for row in board:
        print(" ".join("Q" if c else "." for c in row))
    print()

def is_safe(board, row, col):
    for i in range(row):
        if board[i][col]: return False
    for i,j in zip(range(row-1,-1,-1), range(col-1,-1,-1)):
        if board[i][j]: return False
    for i,j in zip(range(row-1,-1,-1), range(col+1,8)):
        if board[i][j]: return False
    return True

def solve(board, row=0):
    if row == 8:
        print_board(board)
        return
    for col in range(8):
        if is_safe(board, row, col):
            board[row][col] = 1
            solve(board, row+1)
            board[row][col] = 0

board = [[0]*8 for _ in range(8)]
solve(board)
```

```
>>>
========================================
Q . . . . . . .
. . . . Q . . .
. . . . . . . Q
. . . . . Q . .
. . Q . . . . .
. . . . . . Q .
. Q . . . . . .
. . . Q . . . .

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

RESULT

The program successfully solved the **8-Queens Problem** using backtracking and generated all **92 possible solutions**.