

EX.NO:1
Reg.no:220701060

DATE:6/9/2024

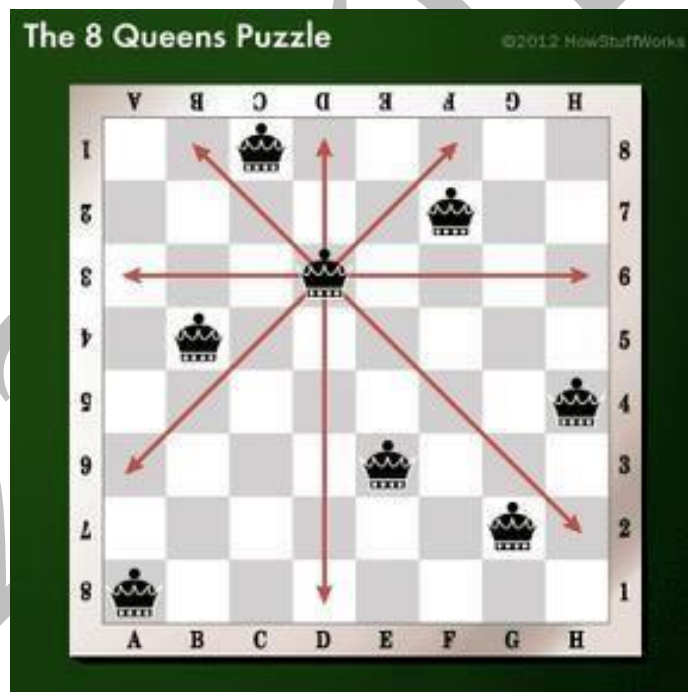
8-QUEENS PROBLEM

AIM:

To implement an 8-Queens problem using Python.

You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other queen on the chessboard. A queen can only be attacked if it lies on the same row, same column, or the same diagonal as any other queen. Print all the possible configurations.

To solve this problem, we will make use of the Backtracking algorithm. The backtracking algorithm, in general checks all possible configurations and test whether the required result is obtained or not. For the given problem, we will explore all possible positions the queens can be relatively placed at. The solution will be correct when the number of placed queens = 8.



CODE:

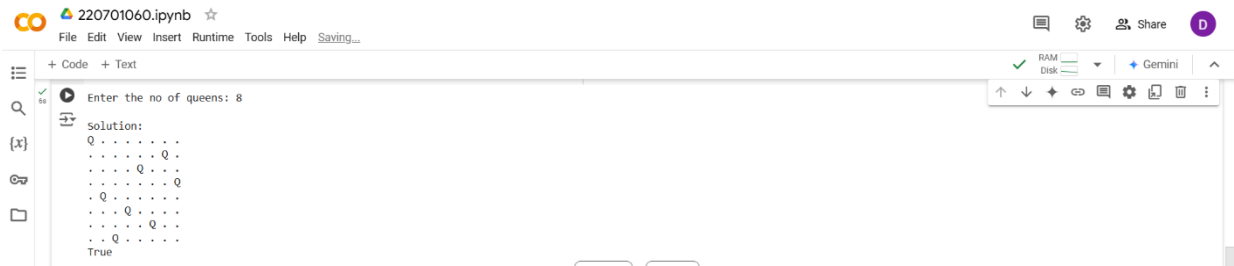
```
def solve_n_queens_util(board, col, n):
    if col >= n:
        return True
    for i in range(n):
        if is_safe(board, i, col, n):
            board[i][col] = 1
            if solve_n_queens_util(board, col + 1, n):
                return True
            board[i][col] = 0
    return False

def print_board(board, n):
    print("\nSolution:")
    for row in board:
        for cell in row:
            if cell == 1:
                print('Q', end=' ')
            else:
                print('.', end=' ')
        print()

def solve_n_queens(n):
    board = [[0 for _ in range(n)] for _ in range(n)]
    if not solve_n_queens_util(board, 0, n):
        print("No solution exists")
        return False
    print_board(board, n)
    return True

n = int(input("Enter the no of queens: "))
solve_n_queens(n)
```

OUTPUT:



The screenshot shows a Jupyter Notebook window with the title "220701060.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for saving, running, and other functions. The notebook has two tabs: "Code" and "Text". The "Code" tab is active, displaying the following code and output:

```
Enter the no of queens: 8
```

The output shows the solution to the 8-Queens problem, represented as an 8x8 chessboard with 'Q' indicating the position of a queen in each row. The solution is:

```
solution:
Q . . . . .
. . . . Q .
. . . Q . .
. . . . . Q
. Q . . . .
. . . . . Q
. . . . . Q
. Q . . . .
True
```

The chessboard is an 8x8 grid. The first row has a queen at column 1. The second row has a queen at column 8. The third row has a queen at column 4. The fourth row has a queen at column 7. The fifth row has a queen at column 2. The sixth row has a queen at column 7. The seventh row has a queen at column 7. The eighth row has a queen at column 2. The output ends with "True".

RESULT:

Thus, the 8-Queens program has been implemented successfully.