

EX.NO:01

DATE:4.9.2024

Reg.no:220701015

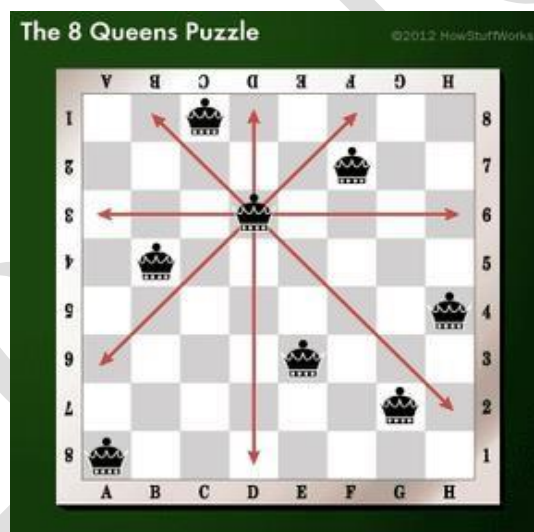
N- QUEENS PROBLEM

AIM:

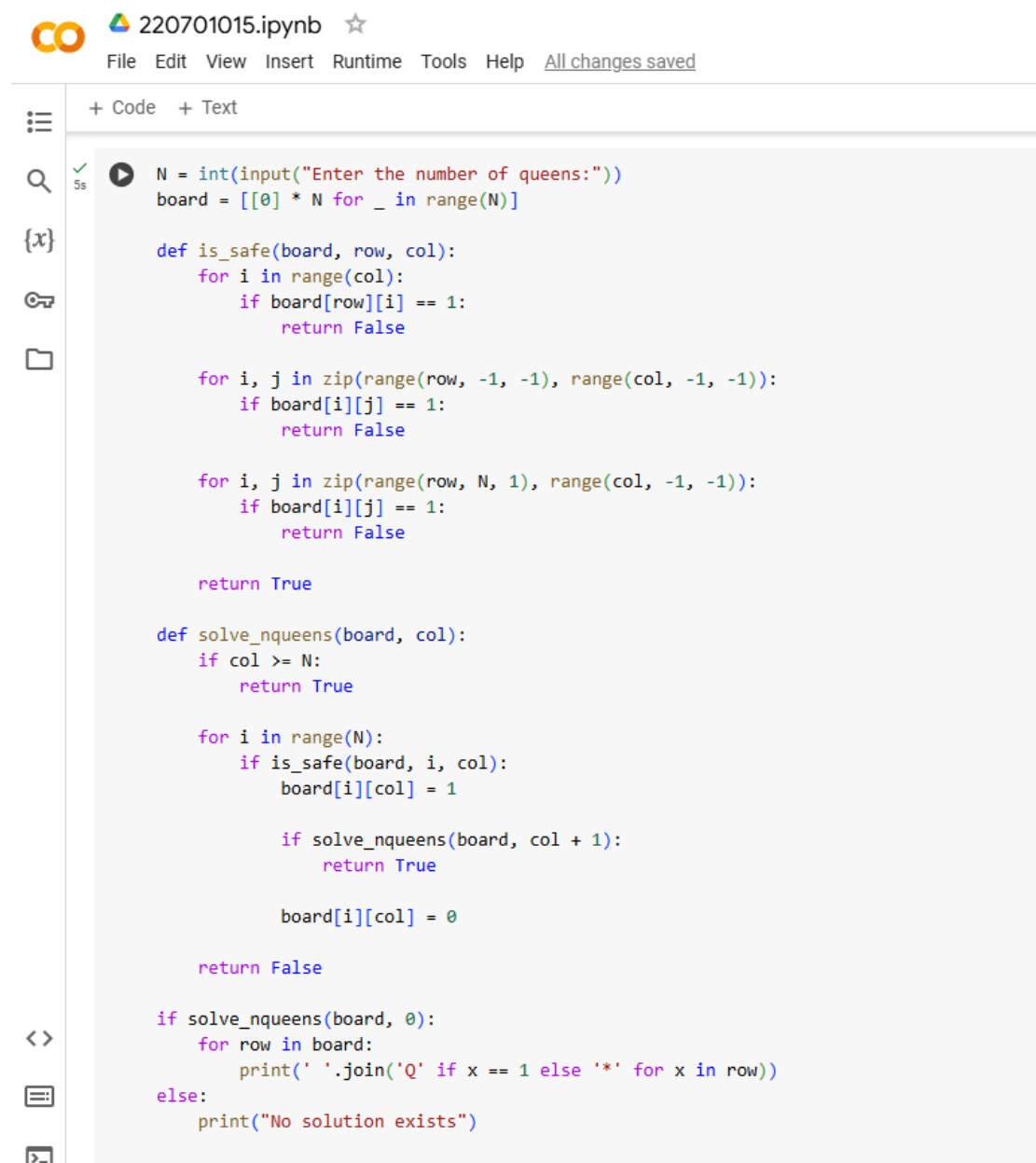
To implement an 8-Queens problem using Python.

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:



The image shows a Jupyter Notebook interface with a file named '220701015.ipynb'. The code defines a function 'is_safe' to check if a queen can be placed at a given position on the board. It then defines a recursive function 'solve_nqueens' to find all possible solutions. The code uses a board represented as a list of lists, where 1 indicates a queen and 0 indicates an empty space. The code is as follows:

```
N = int(input("Enter the number of queens:"))
board = [[0] * N for _ in range(N)]

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

    return True

def solve_nqueens(board, col):
    if col >= N:
        return True

    for i in range(N):
        if is_safe(board, i, col):
            board[i][col] = 1

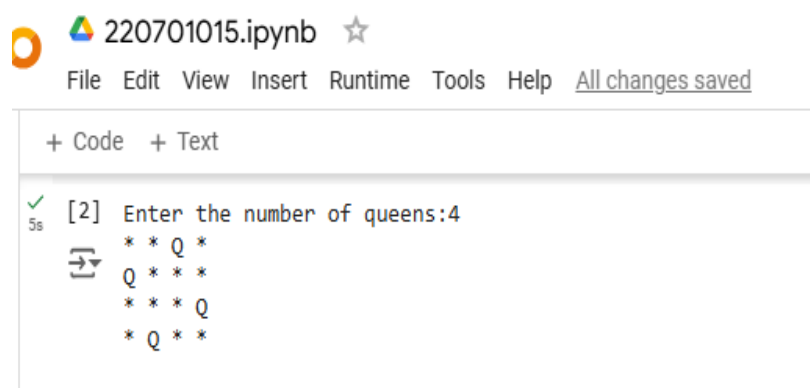
            if solve_nqueens(board, col + 1):
                return True

            board[i][col] = 0

    return False

if solve_nqueens(board, 0):
    for row in board:
        print(' '.join('Q' if x == 1 else '*' for x in row))
else:
    print("No solution exists")
```

OUTPUT:



The image shows the output of the Jupyter Notebook. The user entered '4' for the number of queens. The output displays a 4x4 board with queens placed at positions (0,1), (1,3), (2,2), and (3,0). The board is represented as a list of lists, where 1 indicates a queen and 0 indicates an empty space. The output is as follows:

```
[2] Enter the number of queens:4
* * Q *
Q * * *
* * * Q
* Q * *
```

220701015

220701015