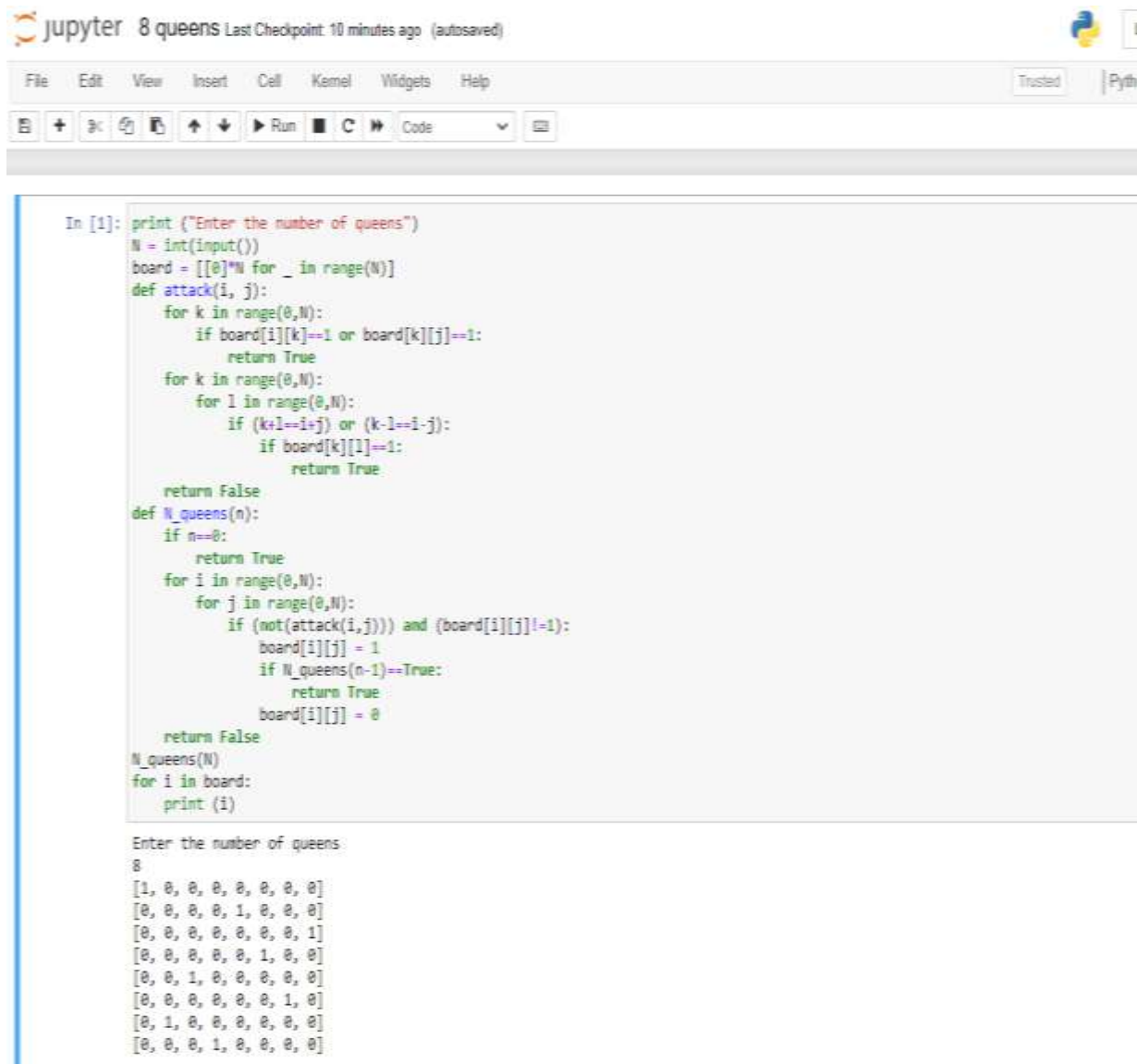


# 8 QUEENS PROBLEM

**AIM :** The 8 queens problem is a problem in which we figure out a way to put 8 queens on an 8x8 chessboard in such a way that no queen should attack the other.



The image shows a Jupyter Notebook interface with the title "8 queens". The notebook contains a Python script to solve the 8 Queens Problem. The script defines a function `attack(i, j)` to check if a queen at (i, j) attacks any other queens on the board. It also defines a recursive function `N_queens(n)` to place n queens on an n x n board. The script prompts the user to enter the number of queens (8) and prints the resulting 8x8 board configuration.

```
In [1]: print ("Enter the number of queens")
N = int(input())
board = [[0]*N for _ in range(N)]
def attack(i, j):
    for k in range(0,N):
        if board[i][k]==1 or board[k][j]==1:
            return True
    for k in range(0,N):
        for l in range(0,N):
            if (k+l==i+j) or (k-l==i-j):
                if board[k][l]==1:
                    return True
    return False
def N_queens(n):
    if n==0:
        return True
    for i in range(0,N):
        for j in range(0,N):
            if (not(attack(i,j))) and (board[i][j]!=1):
                board[i][j] = 1
                if N_queens(n-1)==True:
                    return True
                board[i][j] = 0
    return False
N_queens(N)
for i in board:
    print (i)
```

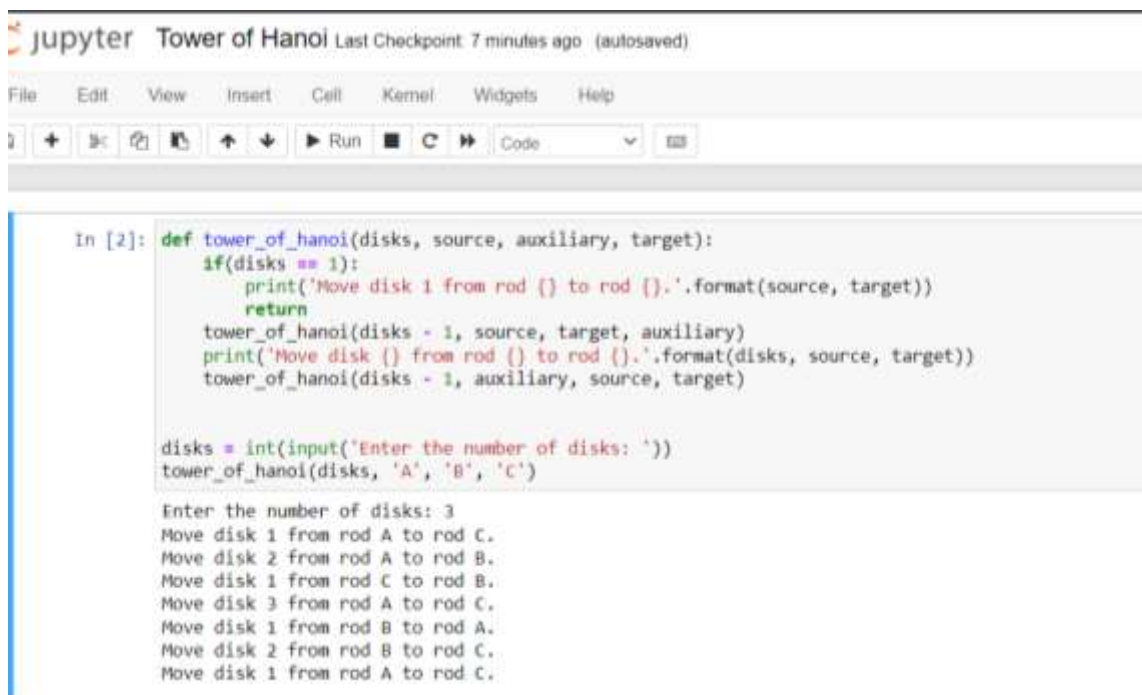
Enter the number of queens  
8  
[1, 0, 0, 0, 0, 0, 0, 0]  
[0, 0, 0, 0, 1, 0, 0, 0]  
[0, 0, 0, 0, 0, 0, 0, 1]  
[0, 0, 0, 0, 0, 1, 0, 0]  
[0, 0, 1, 0, 0, 0, 0, 0]  
[0, 0, 0, 0, 0, 0, 0, 1]  
[0, 1, 0, 0, 0, 0, 0, 0]  
[0, 0, 0, 1, 0, 0, 0, 0]

**RESULT:** 8 queens problem successfully executed

# TOWERS OF HANOI

**AIM:** Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- 1) Only one disk can be moved at a time.
- 2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- 3) No disk may be placed on top of a smaller disk.



```
In [2]: def tower_of_hanoi(disks, source, auxiliary, target):
        if(disks == 1):
            print('Move disk 1 from rod {} to rod {}'.format(source, target))
            return
        tower_of_hanoi(disks - 1, source, target, auxiliary)
        print('Move disk {} from rod {} to rod {}'.format(disks, source, target))
        tower_of_hanoi(disks - 1, auxiliary, source, target)

        disks = int(input('Enter the number of disks: '))
        tower_of_hanoi(disks, 'A', 'B', 'C')

Enter the number of disks: 3
Move disk 1 from rod A to rod C.
Move disk 2 from rod A to rod B.
Move disk 1 from rod C to rod B.
Move disk 3 from rod A to rod C.
Move disk 1 from rod B to rod A.
Move disk 2 from rod B to rod C.
Move disk 1 from rod A to rod C.
```

**RESULT:** Towers of Hanoi problem is successfully executed