Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

Department of
Artificial Intelligence and Data Science

Name: Siddhesh Dilip Khairnar

Class: TY | Division: B | Roll No: 372028

Semester: V | Academic Year: 2023-2024

Subject Name & Code: ADUA31201: Artificial Intelligence

Title of Assignment: Given an input n, print a n X n matrix consisting of numbers from 1 to n each appearing exactly once in each row and each column (Constraint satisfaction problem)

Date of Performance: 06-09-2023 | Date of Submission: 13-09-2023

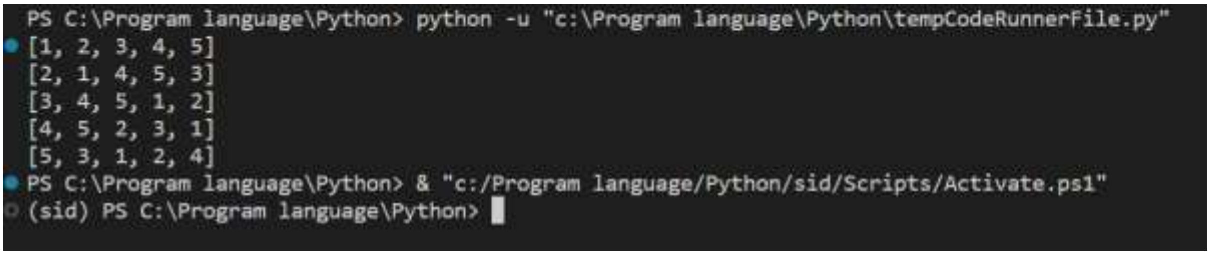
ASSIGNMENT NO. 3

CODE:

```
def is_valid(matrix, row, col, num):
    # Check if 'num' is not already present in the same row or column
    for i in range(len(matrix)):
        if matrix[row][i] == num or matrix[i][col] == num:
            return False
    return True


def solve_matrix(matrix, n, row, col):
    if row == n:
        # All rows are filled, we have a solution
        return True

    # Try placing numbers from 1 to n in the current cell
    for num in range(1, n + 1):
        if is_valid(matrix, row, col, num):
            matrix[row][col] = num

            # Try to fill the next cell in the current row
            if col + 1 < n:
                if solve_matrix(matrix, n, row, col + 1):
                    return True
            else:
                # Move to the next row
                if solve_matrix(matrix, n, row + 1, 0):
                    return True

            # If placing 'num' in the current cell does not lead to a
solution,
            # backtrack and try the next number
            matrix[row][col] = 0

    return False


def generate_magic_matrix(n):
    # Initialize an empty n x n matrix
    matrix = [[0] * n for _ in range(n)]

    # Start solving the matrix
    if solve_matrix(matrix, n, 0, 0):
        return matrix
    else:
        return None


# Example usage:
```

```
n = 5  # Change n to the desired size of the matrix
magic_matrix = generate_magic_matrix(n)

if magic_matrix:
    for row in magic_matrix:
        print(row)
else:
    print("No solution exists for the given n.")
```

OUTPUT:



```
PS C:\Program language\Python> python -u "c:\Program language\Python\tempCodeRunnerFile.py"
[1, 2, 3, 4, 5]
[2, 1, 4, 5, 3]
[3, 4, 5, 1, 2]
[4, 5, 2, 3, 1]
[5, 3, 1, 2, 4]
PS C:\Program language\Python> & "c:/Program language/Python/sid/Scripts/Activate.ps1"
(sid) PS C:\Program language\Python>
```