



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

18CSC305J – ARTIFICIAL INTELLIGENCE LAB

Exp-3: Latin Square Problem

Submitted by-

Name:- Anannya P. Neog

Reg. No. :- RA1911003010367

Course :- Btech

Section :- F1

Branch:- Computer Science Engineering

Sem:- 6th Sem

AI LAB Ex – 3:- Latin Square Problem

Team Members:

- ✓ Richa - 357
- ✓ Anannya - 367
- ✓ Pushan - 371
- ✓ Ankit - 372
- ✓ Tanay - 377

Aim:

To perform Latin Square Problem using Constraint Satisfaction Problem

Objective:

A Latin Square is a $n \times n$ grid filled by n distinct numbers each appearing exactly once in each row and column. Given an input n , we have to print a $n \times n$ matrix consisting of numbers from 1 to n each appearing exactly once in each row and each column.

Procedure:

1. In the first row, the numbers are stored from 1 to n serially.
2. In the second row, the numbers are shifted to the right by one column. i.e, 1 is stored at 2nd column now and so on.
3. In the third row, the numbers are shifted to the right by two columns. i.e, 1 is stored at 3rd column now and so on.
4. We continue the same way for the remaining rows.

Note: There may be more than one possible configuration of a $n \times n$ Latin square.

Code:

```
def printLatin(n):  
  
    k = n + 1  
  
    for i in range(1, n + 1, 1):  
  
        temp = k  
        while (temp <= n) :  
            print(temp, end = " ")  
            temp += 1  
        for j in range(1, k):  
            print(j, end = " ")  
  
        k -= 1  
        print()  
  
n = int(input("enter no"))  
  
printLatin(n)
```

```
1 def printLatin(n):
2     k = n + 1
3
4
5
6     for i in range(1, n + 1, 1):
7         temp = k
8         while (temp <= n):
9             print(temp, end = " ")
10            temp += 1
11            for j in range(1, k):
12                print(j, end = " ")
13            k -= 1
14            print()
15
16
17
18
19     n = int(input("enter no"))
20
21
22 printLatin(n)
23
```

Output:

```
enter no5
1 2 3 4 5
5 1 2 3 4
4 5 1 2 3
3 4 5 1 2
2 3 4 5 1
```

Process exited with code: 0