# CS-594: Python Laboratory

## Instructors: Dr. Sanasam Ranbir Singh and Prof. S.V. Rao

### Topic - Recursion

---

## 1 . Print all the possible subsequences of the String

**Example 1:**
**Input:** str = "abc"
**Output:** a ab abc ac b bc c
**Explanation:** Printing all the 7 subsequence for the string "abc".

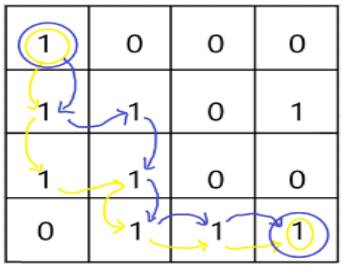
## 2. Rat in a Maze

Consider a rat placed at **(0, 0)** in a square matrix of order **N * N**. It has to reach the destination at **(N – 1, N – 1)**. Find all possible paths that the rat can take to reach from source to destination. The directions in which the rat can move are **'U'(up)**, **'D'(down)**, **'L' (left)**, **'R' (right)**. Value 0 at a cell in the matrix represents that it is blocked and the rat cannot move to it while value 1 at a cell in the matrix represents that rat can travel through it.

```
Input:
N = 4
m[][] = {{1, 0, 0, 0},
         {1, 1, 0, 1},
         {1, 1, 0, 0},
         {0, 1, 1, 1}}

Output: DDRDRR DRDDRR

Explanation:
```




```
The rat can reach the destination at (3, 3) from (0, 0) by two paths - DRDDRR and
DDRDRR, when printed in sorted order we get DDRDRR DRDDRR.
```

## 3. Sudoku Solver

Given a 9×9 incomplete sudoku, solve it such that it becomes a valid sudoku. Valid sudoku has the following properties.
1. All the rows should be filled with numbers(1 – 9) exactly once.
2. All the columns should be filled with numbers(1 – 9) exactly once.
3. Each 3×3 submatrix should be filled with numbers(1 – 9) exactly once.

## 4. Minimum number of jumps to reach end

Given an array arr[] where each element represents the max number of steps that can be made forward from that index. The task is to find the minimum number of jumps to reach the end of the array starting from index 0. If the end isn't reachable, return -1.
Input: arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}
Output: 3 (1-> 3 -> 9 -> 9)

## 5. Combination Sum – Find all unique combinations :

```
candidates = [10,1,2,7,6,1,5], target = 8
Output:
[[1,1,6],
[1,2,5],
[1,7],
[2,6]]
Explanation: These are the unique combinations whose sum is
equal to target.
```

## 6. Count and Say

```
Input: n = 4
Output: "1211"
Explanation:
countAndSay(1) = "1"
countAndSay(2) = say "1" = one 1 = "11"
countAndSay(3) = say "11" = two 1's = "21"
countAndSay(4) = say "21" = one 2 + one 1 = "12" + "11" = "1211"
```

## 7. Word Search

Given a 2D board of letters and a word. Check if the word exists in the board. The word can be constructed from letters of adjacent cells only. ie - horizontal or vertical neighbors. The same letter cell can not be used more than once.

```
Input: board = {{a,g,b,c},{q,e,e,l},{g,b,k,s}},
word = "geeks"
Output: 1
Explanation: The board is-
a g b c
q e e l
g b k s
The letters which are used to make the
"geeks" are colored.
```