**13/06/24 Lab examination. CSA0676 K.Edukondalu(192365010)**

1. **Remove element.**

def remove\_element(lst, element):

"""

Remove the first occurrence of the element from the list.

If the element is not found, return the original list.

"""

if element in lst:

lst.remove(element)

return lst

if \_\_name\_\_ == "\_\_main\_\_":

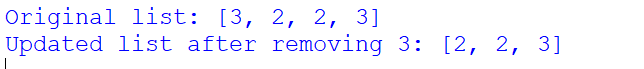
my\_list = [3,2,2,3]

element\_to\_remove = 3

print(f"Original list: {my\_list}")

updated\_list = remove\_element(my\_list, element\_to\_remove)

print(f"Updated list after removing {element\_to\_remove}: {updated\_list}")



1. **Sudoku solver.**

def is\_valid\_move(board, row, col, num):

for i in range(9):

if board[row][i] == num or board[i][col] == num or board[(row//3)\*3 + i//3][(col//3)\*3 + i%3] == num:

return False

return True

def solve\_sudoku(board):

for i in range(9):

for j in range(9):

if board[i][j] == 0:

for num in range(1, 10):

if is\_valid\_move(board, i, j, num):

board[i][j] = num

if solve\_sudoku(board):

return True

board[i][j] = 0

return False

return True

# Example input Sudoku board

input\_board = [

[5, 3, 0, 0, 7, 0, 0, 0, 0],

[6, 0, 0, 1, 9, 5, 0, 0, 0],

[0, 9, 8, 0, 0, 0, 0, 6, 0],

[8, 0, 0, 0, 6, 0, 0, 0, 3],

[4, 0, 0, 8, 0, 3, 0, 0, 1],

[7, 0, 0, 0, 2, 0, 0, 0, 6],

[0, 6, 0, 0, 0, 0, 2, 8, 0],

[0, 0, 0, 4, 1, 9, 0, 0, 5],

[0, 0, 0, 0, 8, 0, 0, 7, 9]

]

if solve\_sudoku(input\_board):

print(True) # Sudoku is solvable

else:

print(False) # Sudoku is not solvable



1. **Count and say.**

from itertools import groupby

def countAndSay(n):

if n == 1:

return "1"

prev = countAndSay(n - 1)

return ''.join(str(len(list(group))) + digit for digit, group in groupby(prev))

# Example Usage

n = 5

result = countAndSay(n)

print(result)



1. **Length of last word.**

def length\_of\_last\_word(s):

words = s.split()

if len(words) == 0:

return 0

return len(words[-1])

# Test the function

input\_string = "Hello World"

result = length\_of\_last\_word(input\_string)

print("Length of the last word in the input string:", result)



1. **Permutation sequences .**

import itertools

n = 3

k = 3

perms = list(itertools.permutations(range(1, n+1), k))

for perm in perms:

print(perm)

