LAB 14

PERMUTATION

def perm(a):

def b(s):

if s == len(a):

result.append(a[:])

for i in range(s, len(a)):

a[s], a[i] = a[i], a[s]

b(s + 1)

a[s], a[i] = a[i], a[s]

result = []

a = list(a)

b(0)

return result

a = input("Enter the Sequence: ")

p = perm(a)

for i in p:

print(''.join(i))

COMBINATION

def c(s, l):

def g(i, p):

if len(p) == l:

r.append(''.join(p) if isinstance(s, str) else p[:])

return

for j in range(i, len(s)):

p.append(s[j])

g(j + 1, p)

p.pop()

r = []

g(0, [])

return r

s = input("Enter sequence: ")

l = int(input("Enter the length: "))

c = c(s, l)

for i in c:

print(i)

instancce

def c(s):

def g(i, p):

r.append(','.join(p) if isinstance(s, str) else p[:])

for j in range(i, len(s)):

p.append(s[j])

g(j + 1, p)

p.pop()

r = []

g(0, [])

return r

s = input("Enter sequence: ")

c = c(s)

for i in c:

print("[",i,"]")

suduko solver

def s(b):

def v(b, r, c, n):

for i in range(9):

if b[r][i] == n or b[i][c] == n:

return False

sr, sc = 3 \* (r // 3), 3 \* (c // 3)

for i in range(sr, sr + 3):

for j in range(sc, sc + 3):

if b[i][j] == n:

return False

return True

def f():

for i in range(9):

for j in range(9):

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

for n in range(1, 10):

if v(b, i, j, n):

b[i][j] = n

if f():

return True

b[i][j] = 0

return False

return True

f()

b = [

[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]]

s(b)

for row in b:

print(row)