Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Experiment 3

Date of Performance : 4 March 2023 **Date of Submission:** 26 March 2023

SAP Id: 60004200132 **Name :** Ayush Jain

Div: B **Batch :** B3

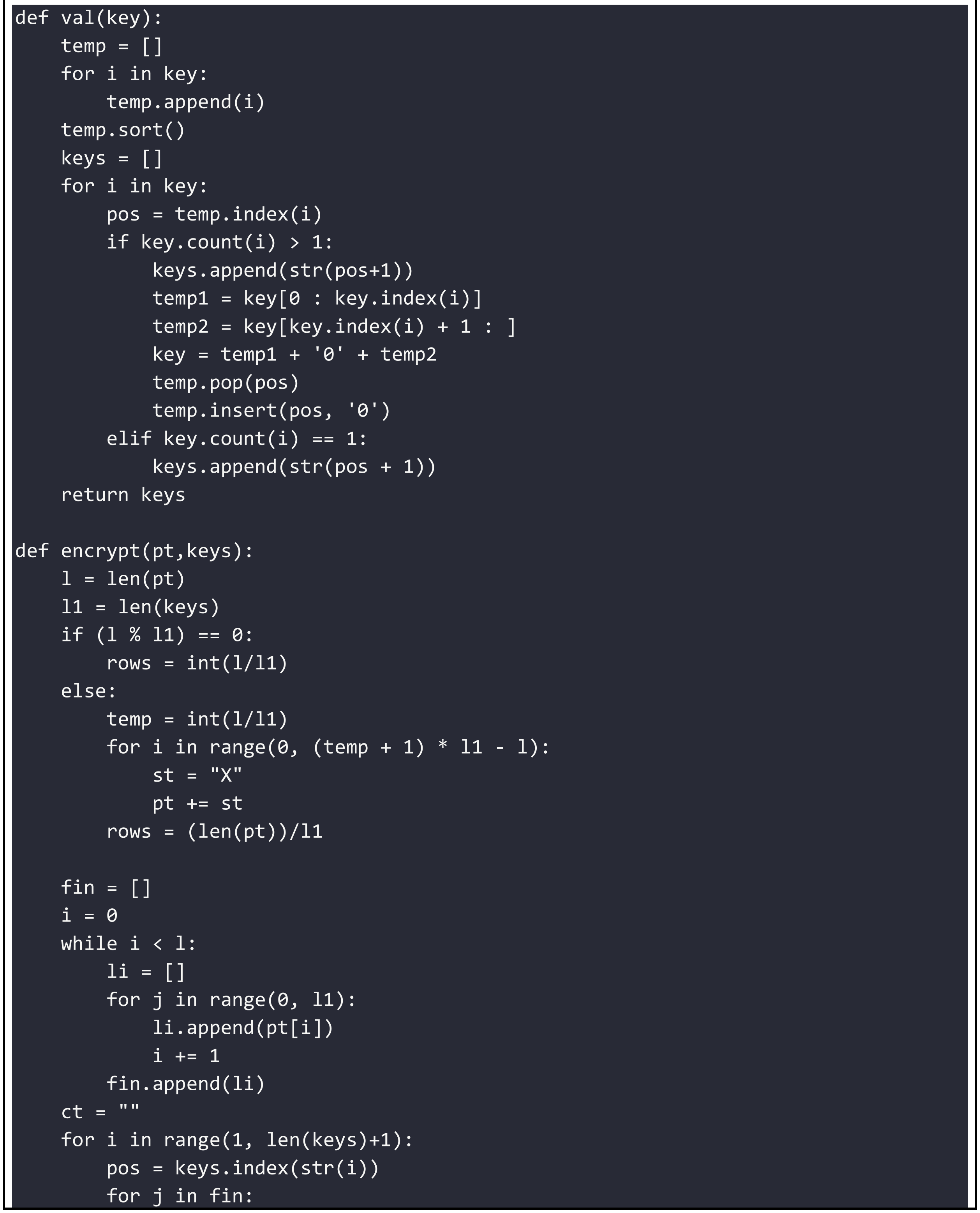
Aim of Experiment

Implement simple columnar transposition technique. The columnar transposition rearranges the plaintext letters, based on a matrix filled with letters in the order determined by the secret keyword.

Theory / Algorithm / Conceptual Description:

The Columnar Transposition Cipher is a type of transposition cipher that rearranges the plaintext letters based on a matrix filled with letters in the order determined by the secret keyword. To encrypt a message using the Columnar Transposition Cipher, we first remove all spaces and convert the plaintext to uppercase. We then determine the number of rows and columns needed for the matrix based on the length of the keyword and the length of the plaintext. We fill the matrix with the plaintext letters row by row, padding with X's if necessary. We then determine the column order based on the keyword, and read the matrix columns in this order to build the ciphertext. To decrypt a message that was encrypted using the Columnar Transposition Cipher, we first determine the number of rows and columns needed for the matrix based on the length of the keyword and the length of the ciphertext. We determine the column order based on the keyword. We then fill the matrix with the ciphertext letters column by column, padding with X's if necessary. We read the matrix rows to build the plaintext message. The security of the Columnar Transposition Cipher relies on the secrecy of the keyword. If the keyword is known or guessed by an attacker, they can easily decrypt the message. However, the Columnar Transposition Cipher can provide a reasonably strong level of security against attackers who do not know the keyword. The ciphertext produced by this cipher can be further strengthened with additional cryptographic techniques such as substitution ciphers and the use of multiple rounds.

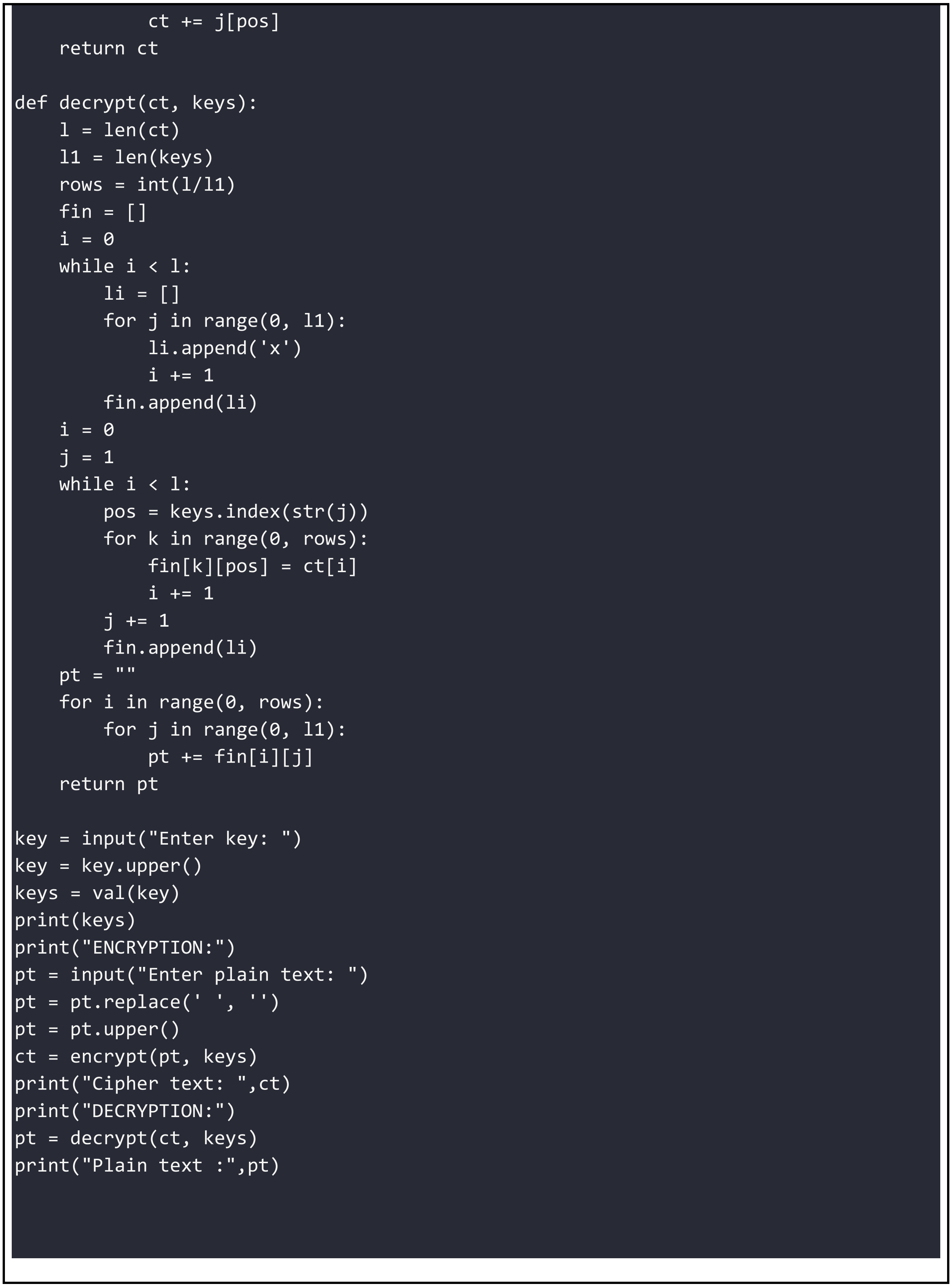
Program



```
def val(key):
    temp = []
    for i in key:
        temp.append(i)
    temp.sort()
    keys = []
    for i in key:
        pos = temp.index(i)
        if key.count(i) > 1:
            keys.append(str(pos+1))
            temp1 = key[0 : key.index(i)]
            temp2 = key[key.index(i) + 1 : ]
            key = temp1 + '0' + temp2
            temp.pop(pos)
            temp.insert(pos, '0')
        elif key.count(i) == 1:
            keys.append(str(pos + 1))
    return keys

def encrypt(pt,keys):
    l = len(pt)
    l1 = len(keys)
    if (l % l1) == 0:
        rows = int(l/l1)
    else:
        temp = int(l/l1)
        for i in range(0, (temp + 1) * l1 - l):
            st = "X"
            pt += st
        rows = (len(pt))/l1

    fin = []
    i = 0
    while i < l:
        li = []
        for j in range(0, l1):
            li.append(pt[i])
            i += 1
        fin.append(li)
    ct = ""
    for i in range(1, len(keys)+1):
        pos = keys.index(str(i))
        for j in fin:
```

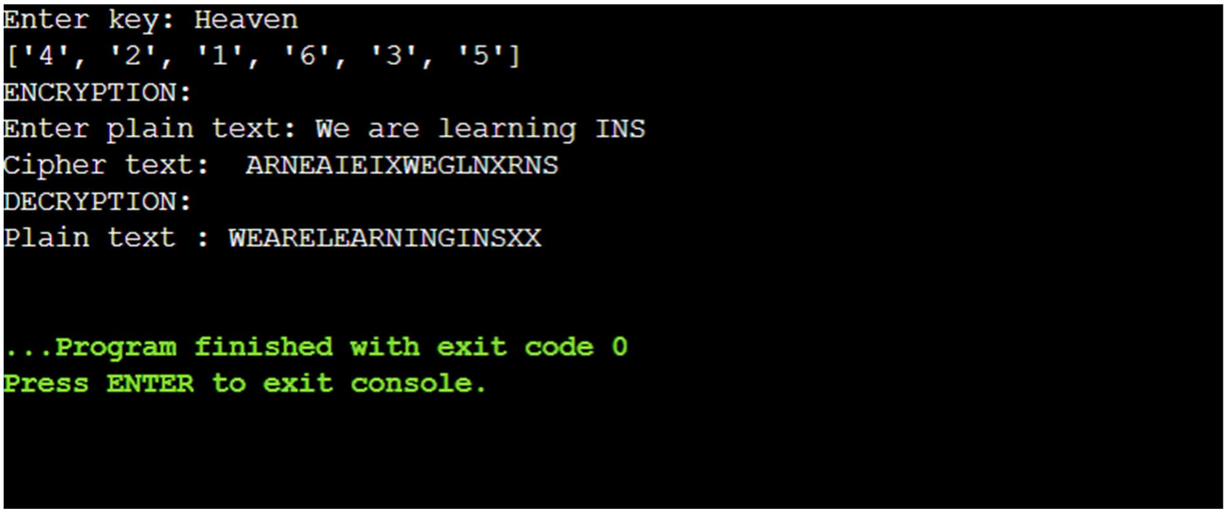


```
            ct += j[pos]
    return ct

def decrypt(ct, keys):
    l = len(ct)
    l1 = len(keys)
    rows = int(l/l1)
    fin = []
    i = 0
    while i < l:
        li = []
        for j in range(0, l1):
            li.append('x')
            i += 1
        fin.append(li)
    i = 0
    j = 1
    while i < l:
        pos = keys.index(str(j))
        for k in range(0, rows):
            fin[k][pos] = ct[i]
            i += 1
        j += 1
        fin.append(li)
    pt = ""
    for i in range(0, rows):
        for j in range(0, l1):
            pt += fin[i][j]
    return pt

key = input("Enter key: ")
key = key.upper()
keys = val(key)
print(keys)
print("ENCRYPTION:")
pt = input("Enter plain text: ")
pt = pt.replace(' ', '')
pt = pt.upper()
ct = encrypt(pt, keys)
print("Cipher text: ",ct)
print("DECRYPTION:")
pt = decrypt(ct, keys)
print("Plain text :",pt)
```

Output



```
Enter key: Heaven
['4', '2', '1', '6', '3', '5']
ENCRYPTION:
Enter plain text: We are learning INS
Cipher text:  ARNEAIEIXWEGLNXRNS
DECRYPTION:
Plain text : WEARELEARNINGINSXX


...Program finished with exit code 0
Press ENTER to exit console.
```