In [1]:

*# Implementation of columnar Transposition cipher.*

In [12]:

**import** **math**

In [23]:

**def** encrypt(key, message):  
 key\_list = []  
 **for** i **in** key:  
 **if** i **not** **in** key\_list:  
 key\_list.append(i)  
  
 mat = [key\_list]  
  
 n = len(key\_list)  
 m = len(message)  
 *# m = 12 n = 5*  
 *# i = 0 , 5, 10*  
 *# i + n = 5, 10, 15*  
 **for** i **in** range(0, m, n):  
 **if** i+n < m:  
 mat.append(message[i:i+n])  
 **else**:  
 li = message[i:]  
 li.extend(['\_']\*(n-len(li)))  
 mat.append(li)  
  
 print(mat)  
  
 key\_list = sorted(key\_list)  
 print(key\_list)  
 cipher\_text = ""  
 **for** i **in** key\_list:  
 index = mat[0].index(i)  
 **for** j **in** range(1, len(mat)):  
 cipher\_text += mat[j][index]  
 *# print(cipher\_text)*  
  
 **return** cipher\_text

In [55]:

**def** decrypt(key, message):  
 key\_list = []  
 **for** i **in** key:  
 **if** i **not** **in** key\_list:  
 key\_list.append(i)  
  
 n = len(key\_list)  
 m = len(message)  
 row = m//n  
  
 dec\_mat = [key\_list]  
 mat = sorted(key\_list)  
   
 **for** i **in** range(row):  
 dec\_mat.append(['']\*n)  
  
 ctr = 0  
  
 **for** i **in** mat:  
 index = dec\_mat[0].index(i)  
 **for** j **in** range(1,row+1):  
 dec\_mat[j][index] = message[ctr]   
 ctr+=1  
 print(dec\_mat)  
 text = ""  
 **for** i **in** range(1,row+1):  
 **for** j **in** range(n):  
 text += dec\_mat[i][j]  
 **return** text

In [54]:

*# key = input()*  
*# message = input()*  
  
key = "heaven"  
message = "ilovemyindia"  
  
print("Key: ", key)  
print("Message: ", message)  
  
  
enc\_msg = encrypt(key, message)  
print("**\n**Encrypted Message: ", enc\_msg)  
  
dec\_msg = decrypt(key, enc\_msg)  
print("**\n**Decrypted Message: ", dec\_msg)

Key: heaven  
Message: ilovemyindia  
[['h', 'e', 'a', 'v', 'n'], ['i', 'l', 'o', 'v', 'e'], ['m', 'y', 'i', 'n', 'd'], ['i', 'a', '\_', '\_', '\_']]  
['a', 'e', 'h', 'n', 'v']  
  
Encrypted Message: oi\_lyaimied\_vn\_  
a 2  
[['h', 'e', 'a', 'v', 'n'], ['', '', 'o', '', ''], ['', '', 'i', '', ''], ['', '', '\_', '', '']]  
e 1  
[['h', 'e', 'a', 'v', 'n'], ['', 'l', 'o', '', ''], ['', 'y', 'i', '', ''], ['', 'a', '\_', '', '']]  
h 0  
[['h', 'e', 'a', 'v', 'n'], ['i', 'l', 'o', '', ''], ['m', 'y', 'i', '', ''], ['i', 'a', '\_', '', '']]  
n 4  
[['h', 'e', 'a', 'v', 'n'], ['i', 'l', 'o', '', 'e'], ['m', 'y', 'i', '', 'd'], ['i', 'a', '\_', '', '\_']]  
v 3  
[['h', 'e', 'a', 'v', 'n'], ['i', 'l', 'o', 'v', 'e'], ['m', 'y', 'i', 'n', 'd'], ['i', 'a', '\_', '\_', '\_']]  
  
Decrypted Message: ilovemyindia\_\_\_

In [ ]:

In [ ]: