Encryption:

```
def convertPlainTextToDiagraphs (plainText):
    for s in range(0,len(plainText)+1,2):
        if s<len(plainText)-1:
            if plainText[s]==plainText[s+1]:
                plainText=plainText[:s+1]+'X'+plainText[s+1:]
    if len(plainText)%2 != 0:
        plainText = plainText[:]+'X'
    return plainText
def generateKeyMatrix (key):
    matrix_5x5 = [[0 for i in range (5)] for j in range(5)]
    simpleKeyArr = []
    for c in key:
        if c not in simpleKeyArr:
            if c == 'J':
                simpleKeyArr.append('I')
            else:
                simpleKeyArr.append(c)
    is_I_exist = "I" in simpleKeyArr
    for i in range(65,91):
        if chr(i) not in simpleKeyArr:
            if i==73 and not is_I_exist:
                simpleKeyArr.append("I")
                is_I_exist = True
            elif i==73 or i==74 and is_I_exist:
                pass
            else:
                simpleKeyArr.append(chr(i))
    index = 0
    for i in range(0,5):
        for j in range(0,5):
            matrix_5x5[i][j] = simpleKeyArr[index]
            index+=1
    return matrix_5x5
def indexLocator (char,cipherKeyMatrix):
    indexOfChar = []
    if char=="J":
        char = "I"
    for i,j in enumerate(cipherKeyMatrix):
        for k,l in enumerate(j):
            if char == l:
                indexOfChar.append(i)
                indexOfChar.append(k)
                return indexOfChar
def encryption (plainText,key):
    cipherText = []
    keyMatrix = generateKeyMatrix(key)
    i = 0
    while i < len(plainText):
        n1 = indexLocator(plainText[i],keyMatrix)
        n2 = indexLocator(plainText[i+1],keyMatrix)
        if n1[1] == n2[1]:
            i1 = (n1[0] + 1) % 5
            j1 = n1[1]
```

```
            i2 = (n2[0] + 1) % 5
            j2 = n2[1]
            cipherText.append(keyMatrix[i1][j1])
            cipherText.append(keyMatrix[i2][j2])
            cipherText.append(", ")
        elif n1[0]==n2[0]:
            i1= n1[0]
            j1= (n1[1] + 1) % 5
            i2= n2[0]
            j2= (n2[1] + 1) % 5
            cipherText.append(keyMatrix[i1][j1])
            cipherText.append(keyMatrix[i2][j2])
            cipherText.append(", ")
        else:
            i1 = n1[0]
            j1 = n1[1]
            i2 = n2[0]
            j2 = n2[1]
            cipherText.append(keyMatrix[i1][j2])
            cipherText.append(keyMatrix[i2][j1])
            cipherText.append(", ")
        i += 2
    return cipherText
def encrypt(text,s):
    result = ""
    for i in range(len(text)):
        char = text[i]
        if (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)
    return result
def main():
    key = input("Enter key: ").replace(" ","").upper()
    text =input("Plain Text: ").replace(" ","").upper()
    s = int(input("Enter no of Rotation: "))
    plainText=encrypt(text,s)
    convertedPlainText = convertPlainTextToDiagraphs(plainText)
    cipherText = " ".join(encryption(convertedPlainText,key))
    print(cipherText)
if __name__ == "__main__":
    main()
```