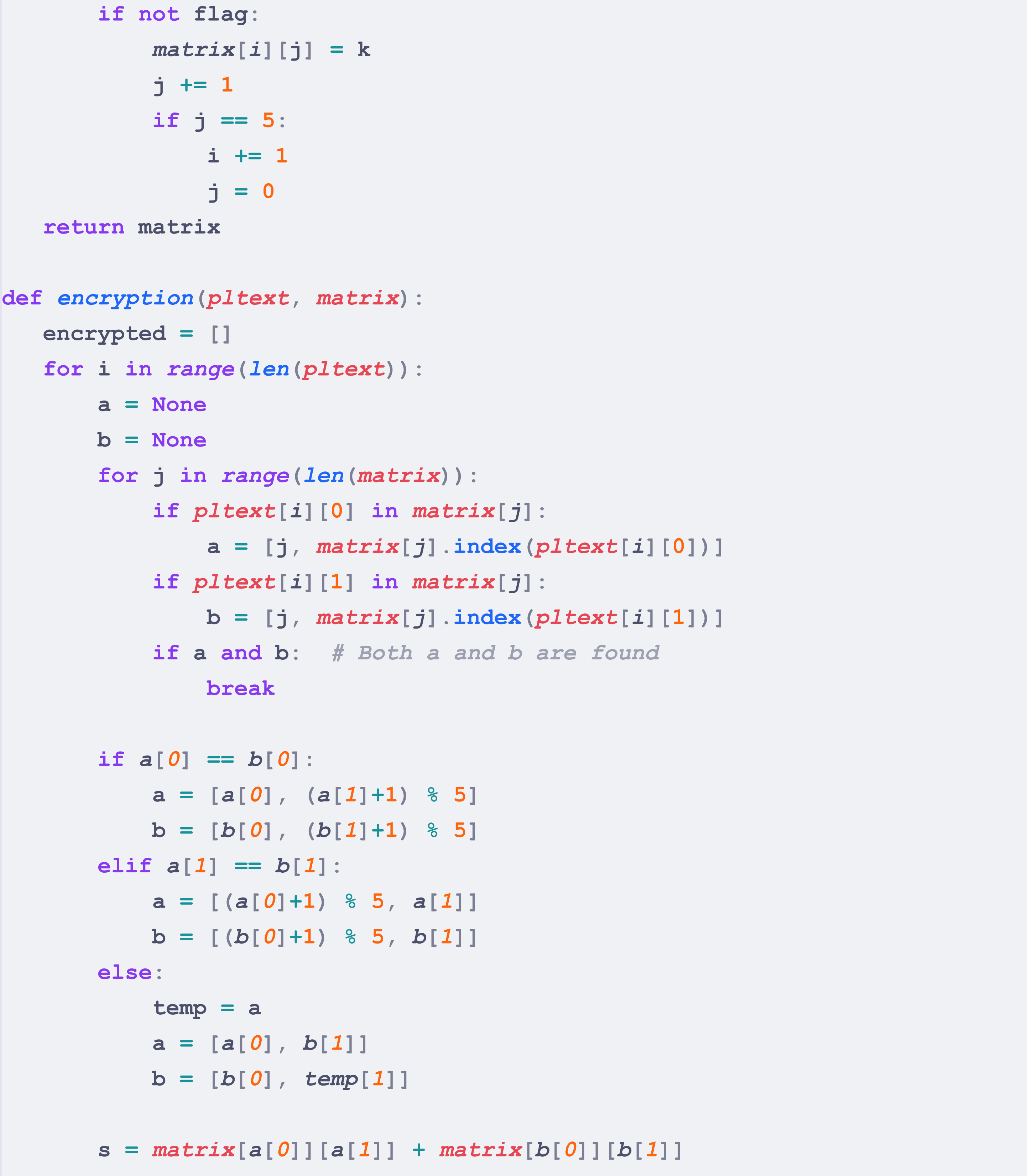
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 61
INS Practical 4

<u>Aim</u> : **Alice wants to send some confidential information to Bob over a secure network. Prepare a key matrix for the given key and apply encryption on the plain text (key is your surname & plain text is your name).**

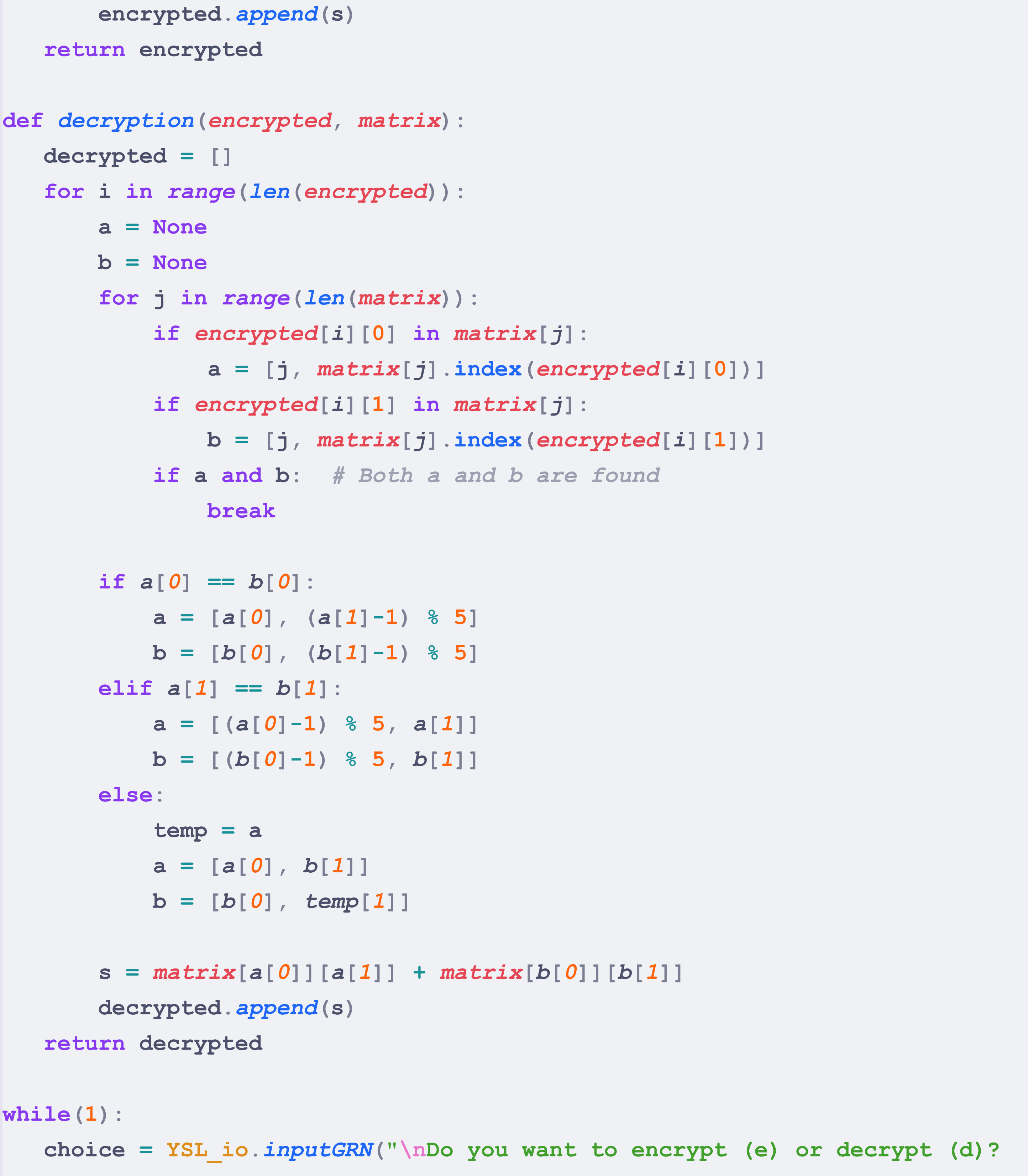
Code :

```
import YSL_io

def getPlainText(plaintext):
    ct = 0
    for i in range(len(plaintext)-1):
        if plaintext[i] == plaintext[i+1]:
            if i % 2 == 0:
                plaintext = plaintext[:i+1] + 'x' + plaintext[i+1:]
    if len(plaintext) % 2 == 1:
        plaintext = plaintext + 'z'
    ct = 0
    n = 2
    pltext = [plaintext[i:i+n] for i in range(0, len(plaintext), n)]
    return pltext

def getMatrix(key):
    str1 = 'abcdefghiklmnopqrstuvwxyz'
    key = key.replace("j", "i")
    matrix = [['' for _ in range(5)] for _ in range(5)]
    i = 0
    j = 0
    for k in key + str1:
        flag = False
        for z in range(5):
            if k in matrix[z]:
                flag = True
                break
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 61
INS Practical 4

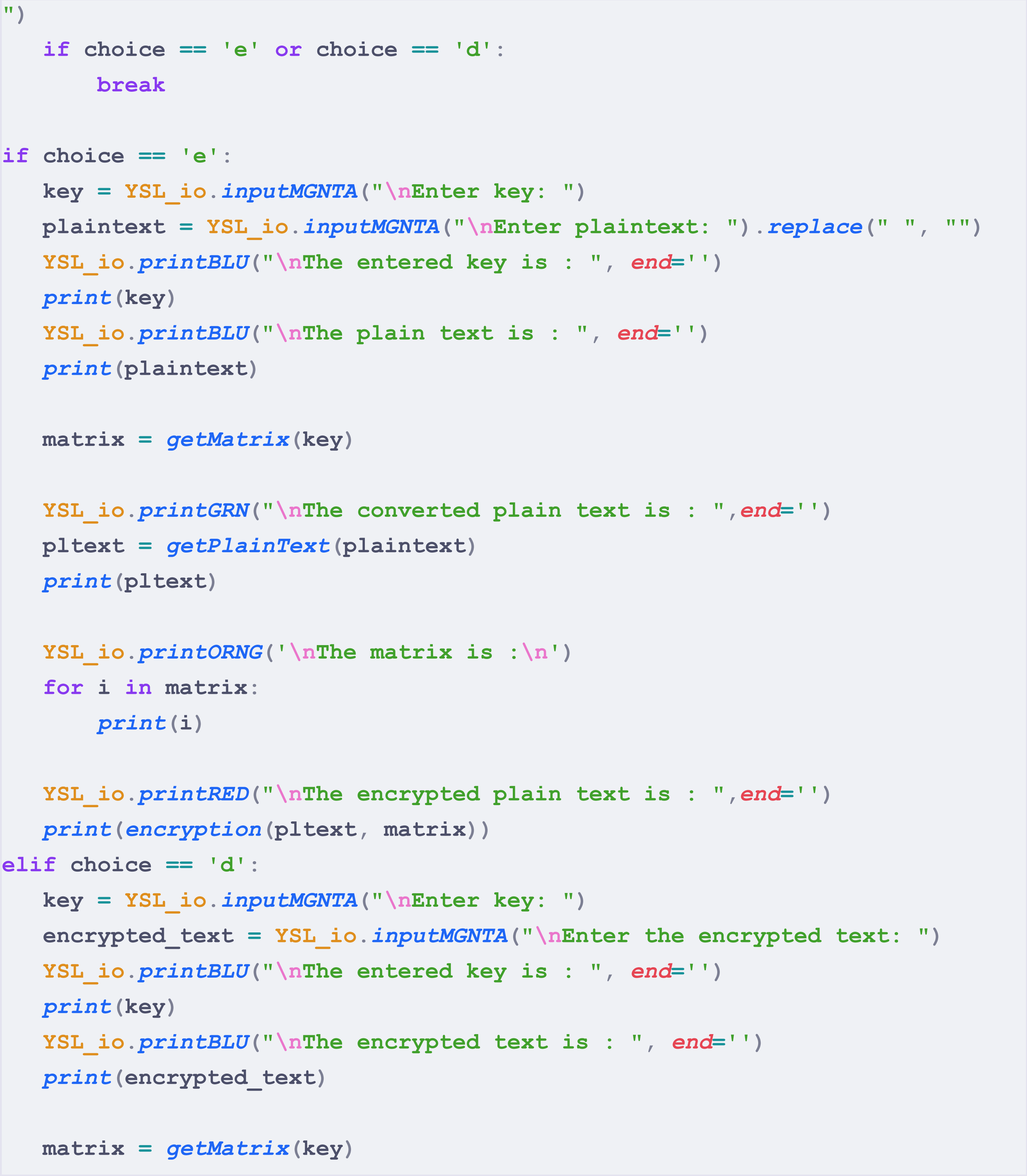


```
        if not flag:
            matrix[i][j] = k
            j += 1
            if j == 5:
                i += 1
                j = 0
    return matrix

def encryption(pltext, matrix):
    encrypted = []
    for i in range(len(pltext)):
        a = None
        b = None
        for j in range(len(matrix)):
            if pltext[i][0] in matrix[j]:
                a = [j, matrix[j].index(pltext[i][0])]
            if pltext[i][1] in matrix[j]:
                b = [j, matrix[j].index(pltext[i][1])]
            if a and b:  # Both a and b are found
                break

        if a[0] == b[0]:
            a = [a[0], (a[1]+1) % 5]
            b = [b[0], (b[1]+1) % 5]
        elif a[1] == b[1]:
            a = [(a[0]+1) % 5, a[1]]
            b = [(b[0]+1) % 5, b[1]]
        else:
            temp = a
            a = [a[0], b[1]]
            b = [b[0], temp[1]]

        s = matrix[a[0]][a[1]] + matrix[b[0]][b[1]]
```



```
        encrypted.append(s)
    return encrypted

def decryption(encrypted, matrix):
    decrypted = []
    for i in range(len(encrypted)):
        a = None
        b = None
        for j in range(len(matrix)):
            if encrypted[i][0] in matrix[j]:
                a = [j, matrix[j].index(encrypted[i][0])]
            if encrypted[i][1] in matrix[j]:
                b = [j, matrix[j].index(encrypted[i][1])]
            if a and b:  # Both a and b are found
                break

        if a[0] == b[0]:
            a = [a[0], (a[1]-1) % 5]
            b = [b[0], (b[1]-1) % 5]
        elif a[1] == b[1]:
            a = [(a[0]-1) % 5, a[1]]
            b = [(b[0]-1) % 5, b[1]]
        else:
            temp = a
            a = [a[0], b[1]]
            b = [b[0], temp[1]]

        s = matrix[a[0]][a[1]] + matrix[b[0]][b[1]]
        decrypted.append(s)
    return decrypted

while(1):
    choice = YSL_io.inputGRN("\nDo you want to encrypt (e) or decrypt (d)?
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 61
INS Practical 4

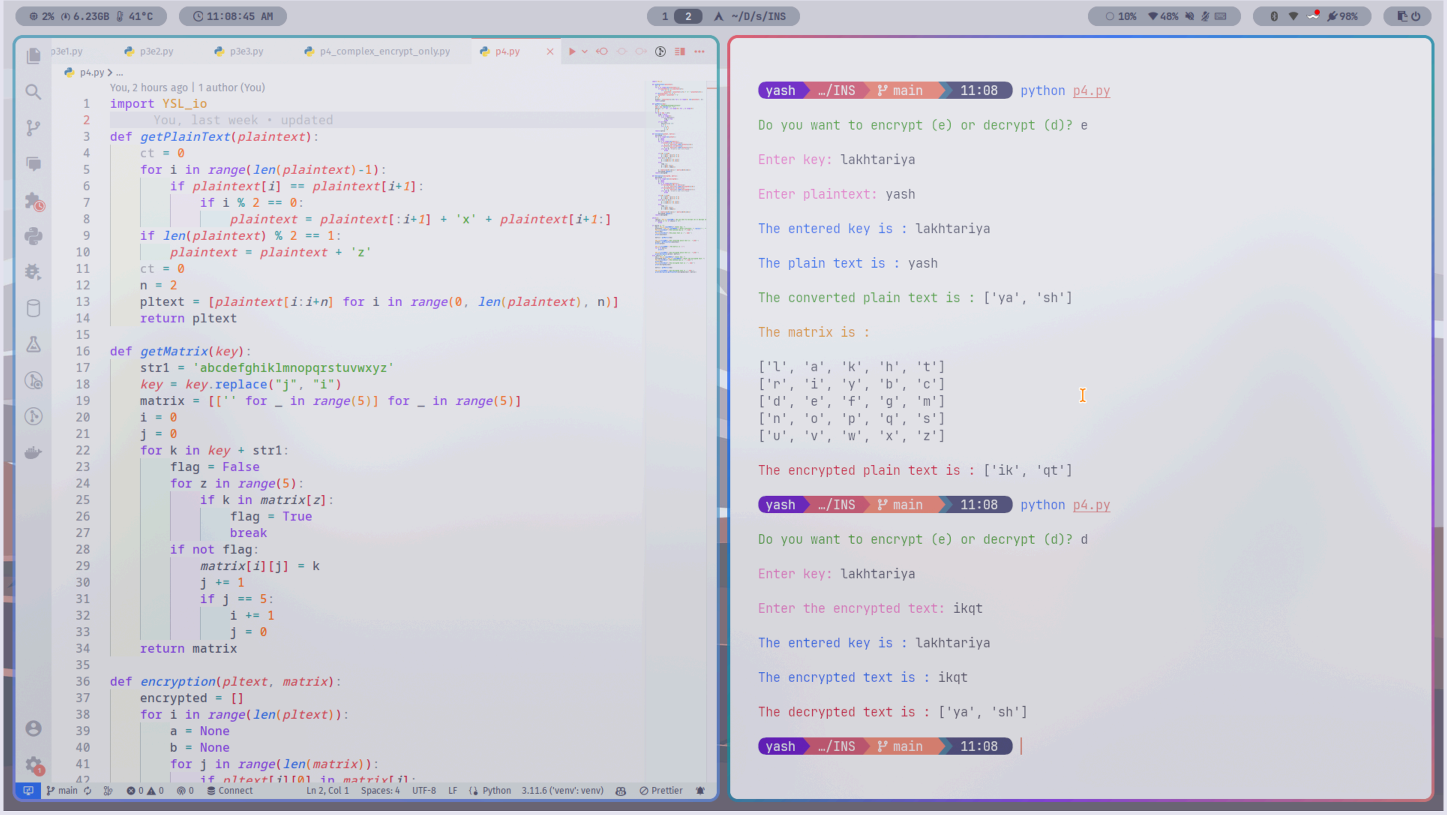


```
")
    if choice == 'e' or choice == 'd':
        break

if choice == 'e':
    key = YSL_io.inputMGNTA("\nEnter key: ")
    plaintext = YSL_io.inputMGNTA("\nEnter plaintext: ").replace(" ", "")
    YSL_io.printBLU("\nThe entered key is : ", end='')
    print(key)
    YSL_io.printBLU("\nThe plain text is : ", end='')
    print(plaintext)

    matrix = getMatrix(key)

    YSL_io.printGRN("\nThe converted plain text is : ",end='')
    pltext = getPlainText(plaintext)
    print(pltext)

    YSL_io.printORNG('\nThe matrix is :\n')
    for i in matrix:
        print(i)

    YSL_io.printRED("\nThe encrypted plain text is : ",end='')
    print(encryption(pltext, matrix))
elif choice == 'd':
    key = YSL_io.inputMGNTA("\nEnter key: ")
    encrypted_text = YSL_io.inputMGNTA("\nEnter the encrypted text: ")
    YSL_io.printBLU("\nThe entered key is : ", end='')
    print(key)
    YSL_io.printBLU("\nThe encrypted text is : ", end='')
    print(encrypted_text)

    matrix = getMatrix(key)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 61
INS Practical 4

```
YSL_io.printRED("\nThe decrypted text is : ",end='')
print(decryption(getPlainText(encrypted_text) , matrix))
```

Output :



```python
import YSL_io

def getPlainText(plaintext):
    ct = 0
    for i in range(len(plaintext)-1):
        if plaintext[i] == plaintext[i+1]:
            if i % 2 == 0:
                plaintext = plaintext[:i+1] + 'x' + plaintext[i+1:]
    if len(plaintext) % 2 == 1:
        plaintext = plaintext + 'z'
    ct = 0
    n = 2
    pltext = [plaintext[i:i+n] for i in range(0, len(plaintext), n)]
    return pltext

def getMatrix(key):
    str1 = 'abcdefghiklmnopqrstuvwxyz'
    key = key.replace("j", "i")
    matrix = [['' for _ in range(5)] for _ in range(5)]
    i = 0
    j = 0
    for k in key + str1:
        flag = False
        for z in range(5):
            if k in matrix[z]:
                flag = True
                break
        if not flag:
            matrix[i][j] = k
            j += 1
            if j == 5:
                i += 1
                j = 0
    return matrix

def encryption(pltext, matrix):
    encrypted = []
    for i in range(len(pltext)):
        a = None
        b = None
        for j in range(len(matrix)):
            if pltext[i][0] in matrix[j]:
```

```
yash  ../INS  main  11:08  python p4.py

Do you want to encrypt (e) or decrypt (d)? e

Enter key: lakhtariya

Enter plaintext: yash

The entered key is : lakhtariya

The plain text is : yash

The converted plain text is : ['ya', 'sh']

The matrix is :

['l', 'a', 'k', 'h', 't']
['r', 'i', 'y', 'b', 'c']
['d', 'e', 'f', 'g', 'm']
['n', 'o', 'p', 'q', 's']
['u', 'v', 'w', 'x', 'z']

The encrypted plain text is : ['ik', 'qt']
yash  ../INS  main  11:08  python p4.py

Do you want to encrypt (e) or decrypt (d)? d

Enter key: lakhtariya

Enter the encrypted text: ikqt

The entered key is : lakhtariya

The encrypted text is : ikqt

The decrypted text is : ['ya', 'sh']

yash  ../INS  main  11:08
```