B. Tech – CE | Semester: VI | Subject: NIS
Lab 1

Additive Cipher : In this encryption algorithm we have limited range anyone attacker can do the all 26 cases and do the crypt analysis and get our key for the decryption so it is not appropriate in sensitive encryption.

For this encryption we add a key value in an ascii of a character and mod it by 26 that is how we do the encryption.

Substitute cipher : In this encryption algorithm there is a substitution between the characters and in this encryption there is 26! Cases that we can make for an encryption.

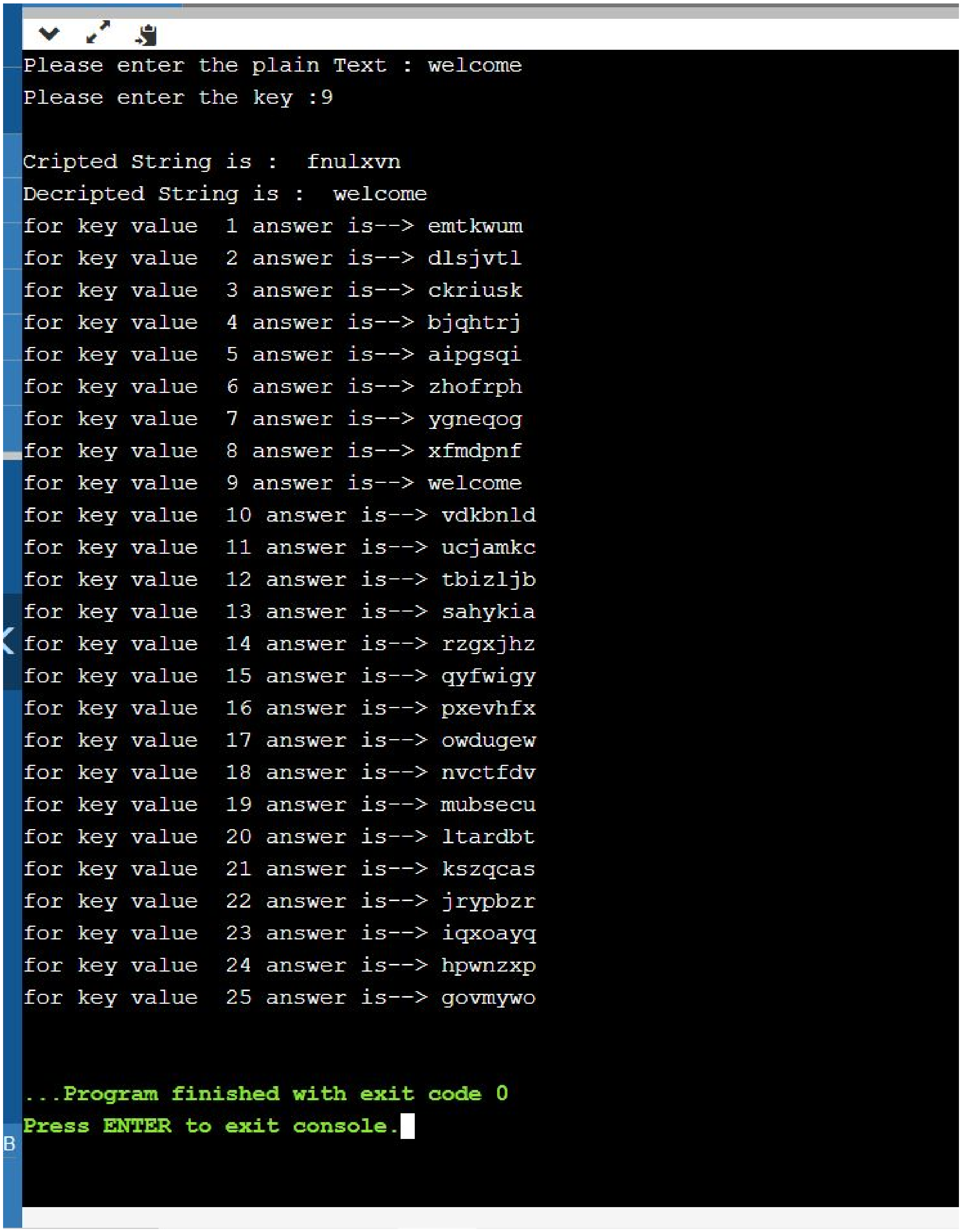
For a → 26 choices ….
For b → 25 choices ….
.
.
.
So on we have..
For z → 1 choice…
So by combining this all cases we can have 26*25*24……*2*1 = 26 !
So to crack this algorithm is very hard for an attacker.

Task 1 : Implement an Additive cipher programme, test it and do a crypt analysis.

Source code : Python

```
#Additive Cypher / Shift Cypher
l=[0]*26
j = 0
count = ord('a')
for i in range(26):
    l[i] = chr(ord('a')+j)
    j+=1

def Cripting (key , Text):
    Text = list(Text)
    Temp = list()
    cripted_text = list()
    for i in Text:
        Temp.append((ord(i)-ord('a')+key)%26)
    for i in Temp:
        cripted_text.append(l[i])
    return "".join(cripted_text)


def Decrypting (key , Text):
    Text = list(Text)
    Temp = list()
    deripted_text = list()
    for i in Text:
        Temp.append((ord(i)-ord('a')-key)%26)
    for i in Temp:
        deripted_text.append(l[i])
    return "".join(deripted_text)

def CryptAnalysis(Cipher):
    Text = list(Cipher)
    for j in range(1 , 26):
        Temp = list()
        for i in Text:
            Temp.append(l[(ord(i)-ord('a')-j)%26])
        print("for key value ",j,"answer is-->","".join(Temp))




if __name__  == "__main__":
    Text = input("Please enter the plain Text : ")
    key = int(input("Please enter the key :"))
    print()
    cripted_text = Cripting( key , Text)
    print("Cryptid String is : ", cripted_text)
    decripted_text = Decrypting(key , cripted_text)
    print("Decrypted String is : " , decripted_text)
    CryptAnalysis(cripted_text)
```

Output :

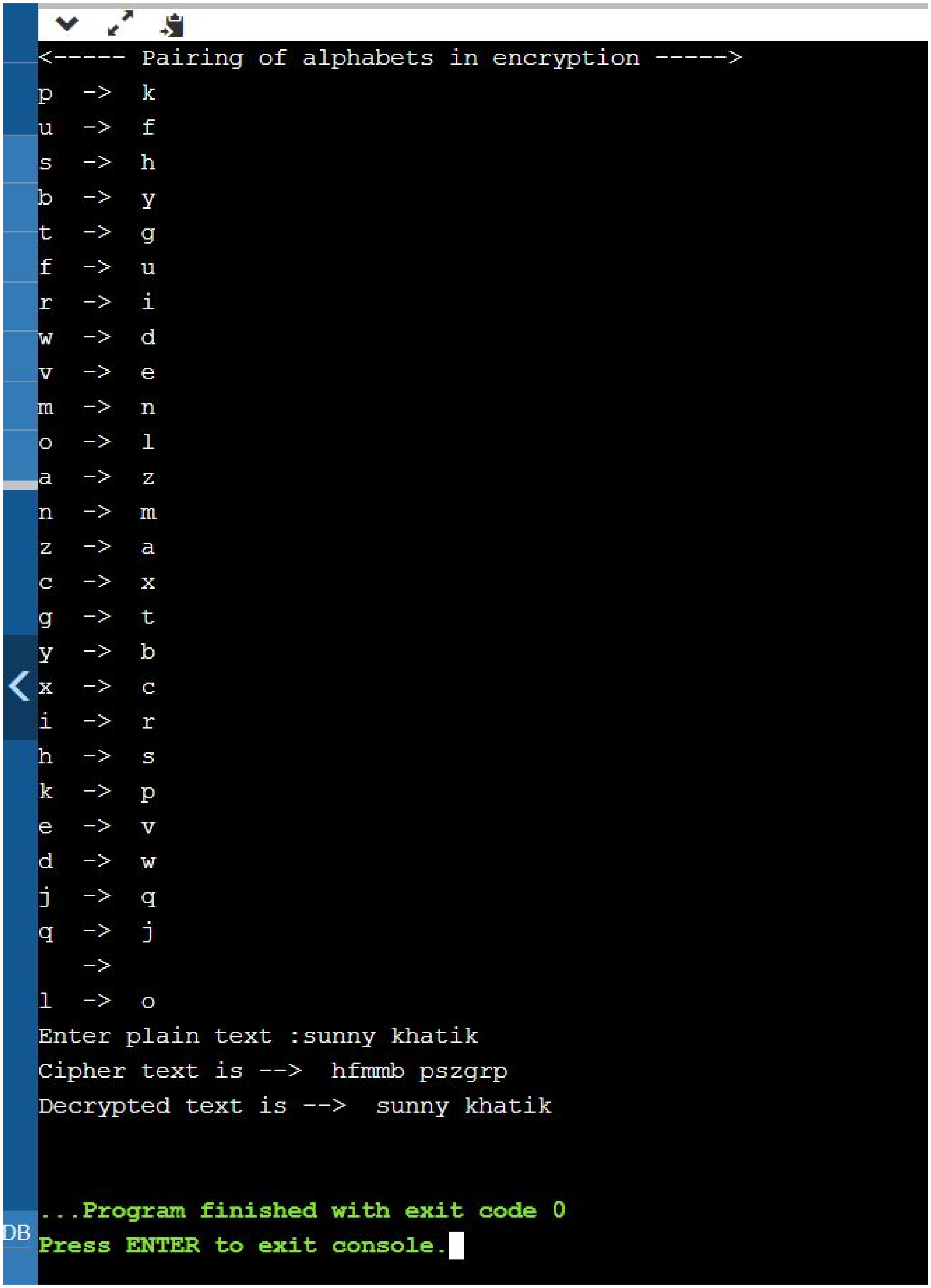

```
Please enter the plain Text : welcome
Please enter the key :9

Cripted String is :  fnulxvn
Decripted String is :  welcome
for key value  1 answer is--> emtkwum
for key value  2 answer is--> dlsjvtl
for key value  3 answer is--> ckriusk
for key value  4 answer is--> bjqhtrj
for key value  5 answer is--> aipgsqi
for key value  6 answer is--> zhofrph
for key value  7 answer is--> ygneqog
for key value  8 answer is--> xfmdpnf
for key value  9 answer is--> welcome
for key value  10 answer is--> vdkbnld
for key value  11 answer is--> ucjamkc
for key value  12 answer is--> tbizljb
for key value  13 answer is--> sahykia
for key value  14 answer is--> rzgxjhz
for key value  15 answer is--> qyfwigy
for key value  16 answer is--> pxevhfx
for key value  17 answer is--> owdugew
for key value  18 answer is--> nvctfdv
for key value  19 answer is--> mubsecu
for key value  20 answer is--> ltardbt
for key value  21 answer is--> kszqcas
for key value  22 answer is--> jrypbzr
for key value  23 answer is--> iqxoayq
for key value  24 answer is--> hpwnzxp
for key value  25 answer is--> govmywo


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

Task 2 : Implement a substitution cipher programme and test it on different test cases.

Source Code :

```python
#Monoaphabatic cipher or substitute cipher.
from collections import defaultdict

d = defaultdict(lambda: " ")
for i in range(26):
    d[chr(97+i)] = chr(122-i)

d[" "]=" "
print("<----- Pairing of alphabets in encryption ----->")
for i in d:
    print( i , " -> ", d[i])

def DoEncryption(text):
    cipher_text=""
    for i in text:
        cipher_text+=d[i]
    return cipher_text

def DoDecryption(cipher_text):
    decrypted_text=""
    for i in cipher_text:
        decrypted_text+=d[i]
    return decrypted_text
if __name__ == "__main__":
    text = input("Enter plain text :")
    cipher_text = DoEncryption(text)
    print("Cipher text is --> ", cipher_text)

    decrypted_text = DoDecryption(cipher_text)
    print("Decrypted text is --> ", decrypted_text)
```

Output :



```
<----- Pairing of alphabets in encryption ----->
p  ->  k
u  ->  f
s  ->  h
b  ->  y
t  ->  g
f  ->  u
r  ->  i
w  ->  d
v  ->  e
m  ->  n
o  ->  l
a  ->  z
n  ->  m
z  ->  a
c  ->  x
g  ->  t
y  ->  b
x  ->  c
i  ->  r
h  ->  s
k  ->  p
e  ->  v
d  ->  w
j  ->  q
q  ->  j
   ->
l  ->  o
Enter plain text :sunny khatik
Cipher text is -->  hfmmb pszgrp
Decrypted text is -->  sunny khatik


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