**Q.no-1, Solution:**

def main():

Text = input("Enter any word: ").lower()

Value = int(input("Enter the value for key: "))

EorD = input(" Enter to encrypt (e) or to decrypt (d): ")

Message = ""

for ch in Text:

if EorD == "e":

Message += chr((ord(ch) - 97 + Value)%26 + 97)

elif EorD =="d":

Message += chr((ord(ch) - 97 - Value)%26 + 97)

print("The given input is:", Text, "to be:", EorD, "with the key value", Value, "To make the new message:", Message)

main()

**Q.no.2, Solution:**

lst= [4, 2, 5, 6]

I= [str(integer) for integer in lst]

a\_string = “ “.join(i)

Res = int(a\_string)

A = 1

sum=res +1

print(sum)

Num = sum

print(“the number after addition is” + str(num))

Res1 = [int(x) for x in str(num)]

print(“the list from number is “ + str(res1))

**Q.no.3, Solution:**

Def add\_binary\_nums(x,y):

Max\_len = max(len(x), len(y))

X = x.zfill(max\_len)

Y=y.fill(max\_len)

Result= ‘ ’

Carryover = 0

For I in range(max\_len-1, -1, -1):

R= carryover

R +=1 if x[i] == ‘1’ else 0

R +=1 if y[i] == ‘1’ else 0

Result = (‘1’ if r%2 == 1 else ‘0’) + result

Carryover = 0 if r<2 else 1

If carryover !=0 : result = ‘1’ + result

Return result.zfill(max\_len)

Print(add\_binary\_nums(‘1011’, ‘10’))

**Q.no.4, Solution:**

Import math

n=eval(input(“the given input is: “))

Def Div(swag):

Sum = 0:

For I in range(1, swag):

If(swag % I == 0):

Sum = sum + I

Return sum:

Def amicable(iteration):

End = 0

For I in range(1, iteration):

CurrentDivs= Div(int(i)):

resultDivs = Div(currentDivs):

If (I == resultDivs and currentDivs != resultDivs):

if(I>n):

print(i):

End = I

Return I

Break

If (end==0):

amicable(iteration\*2)

Def amc(n):

final\_result = amicable(1500):

amc(n)