Que.1

def encode(word, key):

alphabets = 'abcdefghijklmnopqrstuvwxyz';

alphabetsList = list(alphabets);

encodedWord = '';

for i in range(len(word)):

upper = word[i].isupper()

if upper:

alpha = alphabetsList.index(word[i].lower())

encodedWord = encodedWord + alphabetsList[(alpha+key)%len(alphabetsList)].upper()

else:

alpha = alphabetsList.index(word[i].lower())

encodedWord = encodedWord + alphabetsList[(alpha+key)%len(alphabetsList)]

return encodedWord

Que no.2

def addFunc(listNum):

num = 0

for i in range(len(listNum)):

num = num + listNum[i] \* (pow(10,(len(listNum)-1-i)))

return num+1

Que no.3

def sumBinary(num1, num2):

max\_len = max(len(str(num1)), len(str(num2)))

num1 = ''.join(['0' for i in range(max\_len-len(str(num1)))])+str(num1)

num2 = ''.join(['0' for i in range(max\_len-len(str(num2)))])+str(num2)

carry = False

newNum = ''

for i in range(max\_len):

sum = int(num1[len(num1)-i-1]) + int(num2[len(num2)-i-1])

sum = sum + 1 if carry else sum

newNum = ('1' if sum % 2 == 1 else '0') + newNum

carry = False if sum < 2 else True

if carry:

newNum = '1' + newNum

return newNum

Que no.4

def amicable(num):

num += 1

while num != 0:

m = sum\_divisors(num)

if m != num and sum\_divisors(m) == num:

return num

num = num + 1

def sum\_divisors(num):

divisor = 1

total = 0

while divisor < num:

if num % divisor == 0:

total = total + divisor

divisor = divisor + 1

return total

print(amicable(5))