***EMBEDDED SYSTEMS LAB***

***PROBLEM SHEET 1***

PROBLEM 1:

res = 'y'

while res == "y":

num = int(input("ENTER DECIMAL NUMBER: "))

option = int(input("1. BINARY\n2. OCTAL\n3. HEXADECIMAL\nENTER OPTION NUMBER: "))

if option == 1:

print(bin(num)[2:])

elif option == 2:

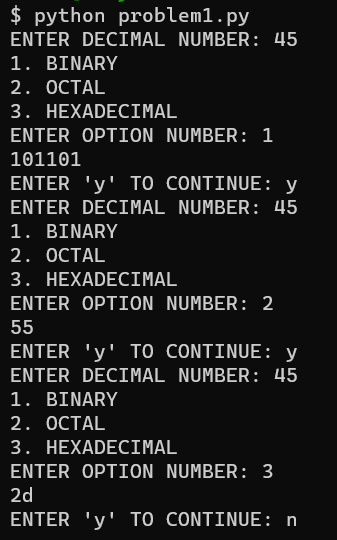
print(oct(num)[2:])

elif option == 3:

print(hex(num)[2:])

res = input("ENTER 'y' TO CONTINUE: ")

OUTPUT:



PROBLEM 2:

res = "y"

while res == "y":

stra = input("ENTER BIT STRING 1: ")

strb = input("ENTER BIT STRING 2: ")

if len(stra) < len(strb):

stra = "0" \* (len(strb) - len(stra)) + stra

elif len(strb) < len(stra):

strb = "0" \* (len(stra) - len(strb)) + strb

print(stra)

print(strb)

result = ""

option = int(input("1. OR\n2. AND\n3. XOR\n4. NAND\nENTER OPTION NUMBER: "))

if option == 1:

print("or")

for i in range(-len(stra), 0):

if stra[i] == "1" or strb[i] == "1":

result += "1"

else:

result += "0"

print(result)

elif option == 2:

for i in range(-len(stra), 0):

if stra[i] == "0" or strb[i] == "0":

result += "0"

else:

result += "1"

print(result)

elif option == 3:

for i in range(-len(stra), 0):

if stra[i] == strb[i]:

result += "0"

else:

result += "1"

print(result)

elif option == 4:

for i in range(-len(stra), 0):

if stra[i] == "0" or strb[i] == "0":

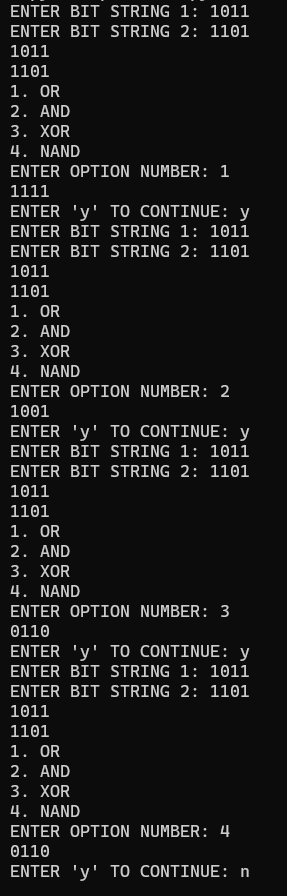
result += "1"

else:

result += "0"

print(result)

OUTPUT:



PROBLEM 3:

res = "y"

while res == "y":

stra = input("ENTER BIT STRING : ")

ressult = ""

option = int(input("1. SHL\n2. SHR\n3. CIL\n4. CIR\nENTER OPTION NUMBER: "))

if option == 1:

result = stra[1:len(stra)] + "0"

elif option == 2:

result = "0" + stra[0:len(stra)-1]

elif option == 3:

result = stra[1:len(stra)] + stra[0]

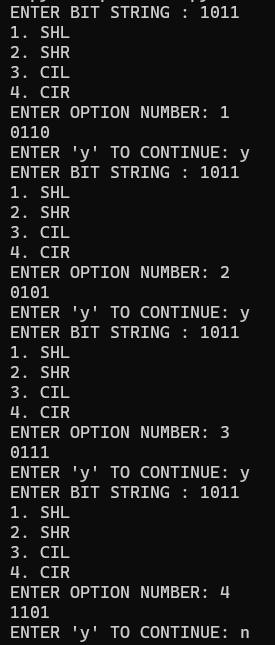
elif option == 4:

result = stra[-1] + stra[0:len(stra)-1]

print(result)

res = input("ENTER 'y' TO CONTINUE: ")

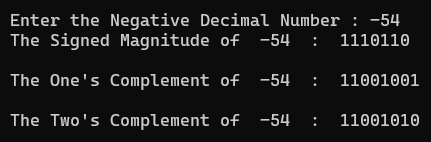
OUTPUT:



PROBLEM 4:

num = int(input('\nEnter the Negative Decimal Number : '))  
  
s = bin(-num)[2:]  
s = '1'+s  
  
print('The Signed Magnitude of ',num,' : ',s)  
  
one = ''  
o = bin(-num)[2:]  
for i in o:  
    if i=='0':  
        one+='1'  
    else:  
        one+='0'  
if len(one)<=8:  
    c = len(one)  
    one = '1'\*(8-len(one))+one  
print('\nThe One\'s Complement of ',num,' : ',one)          
  
two = one  
two = bin(int(two,2)+int('1',2))[2:]  
print('\nThe Two\'s Complement of ',num,' : ',two)

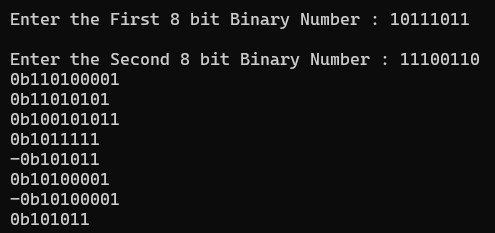
OUTPUT:



PROBLEM 5:

binary1 = input('\nEnter the First 8 bit Binary Number : ')  
binary2 = input('\nEnter the Second 8 bit Binary Number : ')  
nega = ''  
for i in binary1:  
    if i=='1':  
        nega+='0'  
    else:  
        nega+='1'  
negbinary1 = bin(int(nega,2)+int('1',2))  
if len(negbinary1)>8:  
    len1 = len(negbinary1)  
    negbinary1 = negbinary1[len1-8:]  
negb = ''  
for i in binary2:  
    if i=='1':  
        negb+='0'  
    else:  
        negb+='1'  
negbinary2 = bin(int(negb,2)+int('1',2))  
if len(negbinary2)>8:  
    len2 = len(negbinary2)  
    negbinary2 = negbinary2[len2-8:]  
     
bin1addbin2 = bin(int(binary1,2)+int(binary2,2))  
bin1addnegbin2 = bin(int(binary1,2)+int(negbinary2,2))  
negbin1addbin2 = bin(int(negbinary1,2)+int(binary2,2))  
negbin1addnegbin2 = bin(int(negbinary1,2)+int(negbinary2,2))  
bin1subbin2 = bin(int(binary1,2)-int(binary2,2))  
bin1subnegbin2 = bin(int(binary1,2)-int(negbinary2,2))  
negbin1subbin2 = bin(int(negbinary1,2)-int(binary2,2))  
negbin1subnegbin2 = bin(int(negbinary1,2)-int(negbinary2,2))  
print(bin1addbin2)  
print(bin1addnegbin2)  
print(negbin1addbin2)  
print(negbin1addnegbin2)  
print(bin1subbin2)  
print(bin1subnegbin2)  
print(negbin1subbin2)  
print(negbin1subnegbin2)

OUTPUT:



PROBLEM 6:

def setbits(x,pos,num,y):

part = y[len(y)-num:]

d = x[:len(x)-pos] + part + x[len(x)-pos+len(part):]

print('Hence the Modified X is : ',d)

num = int(input('\nEnter the Value of n : '))

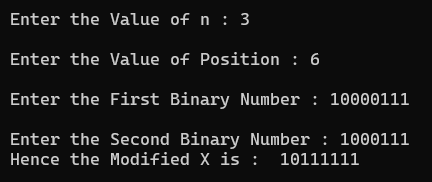
pos = int(input('\nEnter the Value of Position : '))

x = input('\nEnter the First Binary Number : ')

y = input('\nEnter the Second Binary Number : ')

setbits(x,pos,num,y)

OUTPUT:



PROBLEM 7:

import sys, os

def readLastN(file = 'problem7.txt', n = 5):

if os.path.exists(file):

with open(file, 'r') as f:

lines = f.readlines()

if int(n) > len(lines):

return "More lines than in file."

else:

return lines[-int(n)-1: -1]

else:

return "File Does Not Exist."

if len(sys.argv) == 4:

response = readLastN(sys.argv[1], sys.argv[3])

if response == "File Does Not Exist." or response == "More lines than in file.":

print(response)

else:

for i in response:

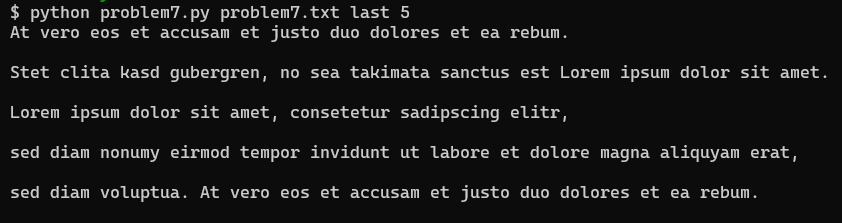
print(i)

else:

for i in readLastN():

print(i)

OUTPUT:



PROBLEM 8:

import random  
num = int(input('\nEnter the value of N so that N random numbers in the range 0.0 - 1.0 to be Generated : '))  
numbers1 = []  
for i in range(num):  
    numbers1.append(random.random())  
numbers1.sort()  
print('\nThe Randomly Generated Sequence from 0.0 - 1.0 containing ',num,' terms is : ',numbers1)  
n = float(input('\nEnter the Upper Bound of the Sequence : '))  
numbers2 = []  
for i in range(num):  
    numbers2.append(random.uniform(0.0,n))  
numbers2.sort()  
print('\nThe Randomly Generated Sequence from 0.0 - ',n,' containing ',num,' terms is : ',numbers2)

OUTPUT:

