1. Write a Python Program to Find LCM?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

# this function takes in a list as an argument

def lcm\_of\_list(l):

r = l.copy()

# let es be an empty set

es = []

maxi = max(l)

while es != r:

es.clear()

for i in r:

if maxi%i == 0:

es.append(i)

continue

if es == r:

break

else:

maxi += 1

print(maxi)

# defining a function to print our required string

try:

lcm\_of\_list([4,5,6,7])

lg.info("Function lcm\_of\_list() has been called")

except exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python Program to Find HCF?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def get\_list\_for\_HCF(\*args):

# we are getting list of elements from user input in function and making a list of those elements in main\_list

l1 = []

main\_list = []

l1.append(args)

for i in l1[0]:

main\_list.append(i)

# we are copyting all elements into anothe list called r

r = main\_list.copy()

# we are useing an Empty Set called es for appending all values to be used later in finding HCF

es = []

count = 1

# this is an empty list called hcf used for appending list of all existing factors involved in the program

hcf = []

main = []

while count <= min(main\_list):

for i in r:

if i%count == 0:

es.append(i)

hcf.append(count)

continue

if i%count != 0:

break

count += 1

for j in hcf:

if hcf.count(j) == len(main\_list):

main.append(j)

print("The HCF of given list of numbers is : " , max(main))

try:

get\_list\_for\_HCF(3,2,60)

lg.info("Function has been called")

except exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

# this function changes your input number to, binary, octal and hexagesimal system.

def get\_bi\_oct\_hex():

number = int(input("Enter the number you want to convert to Binary and Hexadecimal form: "))

num = number

num2 = number

num3 = number

num\_image = number

# defining a few empty sets to be used later on in the program

es = []

es1 = []

es2 = []

es3 = []

# defining the empty strings which will later become the answers

binary = ""

octa = ""

hexa = ""

# this loop

while num > 0:

rem = num%2

es.append(rem)

num = num//2

es.reverse()

for i in es:

binary += str(i)

print("The binary form of", number, "is", binary)

while num3 >0:

tip = num3%8

es3.append(tip)

num3 = num3//8

es3.reverse()

for k in es3:

octa += str(k)

print("The octal form of number is", octa)

while num2 > 0:

jay1 = num2%16

if jay1 in range(10,16):

if jay1 == 10:

es1.append("A")

if jay1 == 11:

es1.append("B")

if jay1 == 12:

es1.append("C")

if jay1 == 13:

es1.append("D")

if jay1 == 14:

es1.append("E")

if jay1 == 15:

es1.append("F")

if jay1 <= 9:

es1.append(jay1)

num2 = num2//16

if num2<10:

es.append(num2)

es1.reverse()

for j in es1:

hexa += str(j)

print("The hexadecimal form of ",num\_image ,"is ", hexa)

try:

get\_bi\_oct\_hex()

lg.info("Function get\_bi\_oct\_hex() has been called")

except exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python Program To Find ASCII value of a character?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

# this function changes your input number to, binary, octal and hexagesimal system.

def get\_ASCII(test\_character):

print("The ASCII value of", test\_character , "is", ord(test\_character))

try:

get\_ASCII("A")

lg.info("Function get\_ASCII() has been called")

except exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass

1. Write a Python Program to Make a Simple Calculator with 4 basic mathematical operations?

Ans.

import logging as lg

# importing logging so every function call of

lg.basicConfig(filename ='C:\\Users\\Home\\Johns python talent\\logging\\testlog1.log', level =lg.INFO , format = '%(asctime)s %(message)s')

def call\_calculator():

while True:

print("For addition of two numbers Enter the value: 1 :\nFor difference of two numbers Enter the option 2: \nFor multiplication of two numbers enter 3\nFor division of Two numbers enter 4\n Any number or character entered other than 1 to 4 will restart program")

user\_option = input("Choose your option: ")

if user\_option == '1' or user\_option == '2' or user\_option == '3' or user\_option =='4':

first\_number = int(input("Enter your first number: "))

second\_number = int(input("Enter your second number: "))

if user\_option == '1':

add = first\_number+second\_number

print("the addition of the two numbers are", add)

if user\_option == '2':

diff = first\_number-second\_number

print("the difference of the two numbers are", diff)

if user\_option == '3':

mult = first\_number\*second\_number

print("the addition of the two numbers are", mult)

if user\_option == '4':

div1 = first\_number/second\_number

div2 = second\_number/first\_number

print("the division of the two numbers in both orientations are", "a/b =", div1 , "and b/a =",div2)

break

else:

print(" Please enter numbers 1 to 4 only: ")

try:

call\_calculator()

lg.info("Function call\_calculator() has been called")

except exception as e:

print("There was an error called: ",e)

else:

pass

finally:

pass