LABCYCLE 1

EXPERIMENT NO:1

LEAP YEAR

Date: 10/11/2022

AIM: Display future leap years from current year to a final year entered by user.

ALGORITHM:

Step 1: Start.

Step 2: Input current year and future year.

Step 3: Repeat step 3 to 7 from current year <=future year.

Step 4: Check whether year%4==0, if true then move to next step. If false then move to step 6.

Step 5: Check whether year%4==100, if true then move to next step. If false then move to step 6.

Step 6: Check whether year%4==0, if true then move to next step. If false then move to step 6.

Step 7: Print leap years.

Step 8: Stop.

SOURCE CODE:

year=int(input("Enter the current year\t"))

fut=int(input("Enter the future year\t"))

print("The leap years in range are\n" )

for year in range(year,fut+1):

if year%4==0 and year%100!=0 or year%400==0:

print(year)

OUTPUT:



RESULT:

Program to display leap entered by user has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:2

LIST OF VALUES

Date: 10/11/2022

Aim: List comprehensions:

(a) Generate positive list of numbers from a given list of integers

(b) Square of N numbers

(c) Form a list of vowels selected from a given word

(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

(a)Generate positive list of numbers from a given list of integers

ALGORITHM:

Step 1: Start.

Step 2: Input a list of positive and negative integers.

Step 3: Repeat step 4 until end of list.

Step 4: If i>0 then append to new list.

[end of loop]

Step 5: Print the new list.

Step 6: Stop.

SOURCE CODE:

list1=[2,3,9,6,-3,-5,10,-15,11]

li=[]

print("The positive numbers in list are\n")

for i in list1:

if i>0:

li.append(i)

print(li)

(b) Square of N numbers

ALGORITHM:

Step 1: Start.

Step 2: Input the number, n.

Step 3: Repeat step 4 with range (1, n+1).

Step 4: Find square, sq=i\*i.

Print(sq).

Step 5: Stop.

SOURCE CODE:

n=int(input("Enter the limit to find the square of N no:\n"))

for i in range(1,n+1):

sq=i\*i

print("The squares of ",i,"is",sq,"\n")

(c)Form a list of vowels selected from a given word

ALGORITHM:

Step 1: Start.

Step 2: Input empty list.

Step 3: Input a word.

Step 4: check if each letter of word present in list of vowels, if true.

append the letter to empty list, goto step 4.

Step 5: Print list.

Step 6: Stop.

SOURCE CODE:

l=[]

wrd=input("Enter a word")

vowels=['a','e','i','o','u','A','E','I','O','U']

for i in wrd:

if i in vowels:

l.append(i)

print(l,"\t")

(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

ALGORITHM:

Step 1: Start.

Step 2: Input the word.

Step 3: Print ordinal value by iterating the word.

Step 4: Stop

SOURCE CODE:

wrd=input("Enter the word\n")

li=[]

for i in wrd:

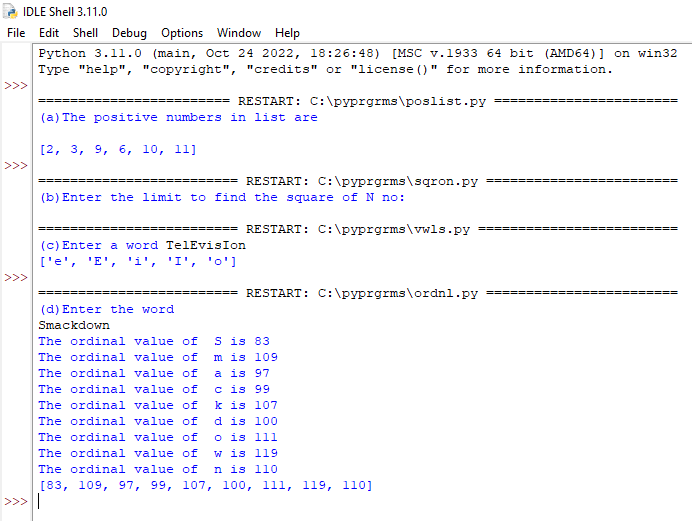
d=ord(i)

print("The ordinal value of ",i,"is",ord(i))

li.append(d)

print(li)

OUTPUT:



RESULT:

Program to perform list operation has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:3

OCCURRENCES OF WORD

Date: 10/11/2022

AIM: Count the occurrences of each word in a line of text.

ALGORITHM

Step 1: Start.

Step 2: Set Define function word\_count(str)

count=dict()

words=str.split()

Take the count of each word.

Print(count).

Step 3: Input a string.

Step 4: Call function, word\_count(str).

Step 5: Stop.

SOURCE CODE:

def word\_count(str):

count=dict()

words=str.split()

for w in words:

if w in count:

count[w]+=1

else:

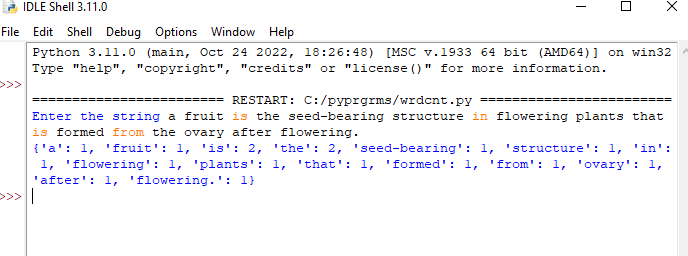
count[w]=1

print(count)

a=input("Enter the string ")

word\_count(a)

OUTPUT:



RESULT:

Program to count the occurrences of each word has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:4

LIST OF INTEGERS

Date: 10/11/2022

AIM: Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

ALGORITHM:

Step 1: Start.

Step 2: Input the two empty list.

Step 3: Input the limit,n.

Step 4: Append each element to the list.

Step 5: Append each element greater than 100 in another list.

Step 6: Print new list.

Step 7: Stop.

SOURCE CODE:

lis=[]

a=[]

n=int(input("Enter the limit of the list \t"))

print("Enter the list of elements")

for i in range(0,n):

print("Enter the element no:-{}:".format(i+1))

elm=int(input())

lis.append(elm)

print("The entered list is ",lis)

for i in lis:

if i>100:

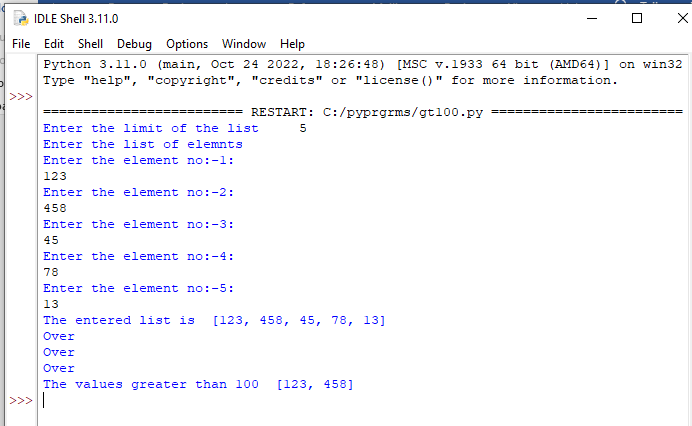
a.append(i)

else:

print("Over")

print("The values greater than 100 ",a)

OUTPUT:



RESULT:

Program to display list of integers has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:5

OCCURRENCES OF “a”

Date: 10/11/2022

AIM: Store a list of first names. Count the occurrences of ‘a’ within the list.

ALGORITHM:

Step 1: Start.

Step 2: Input an empty list and limit of list.

Step 3: Append each element to the list.

Step 4: Initialize count=0.

Step 5: Check the presence of “a” in the list. Update count.

Step 6: Print count.

Step 7: Stop.

SOURCE CODE:

list1=[]

len=int(input("Enter the number of names you want to insert "))

for i in range(0,len):

print("Enter the name ",i+1," you want to insert ")

fname=input()

list1.append(fname)

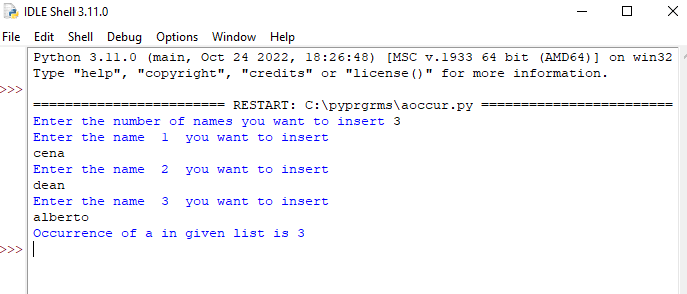
count\_a=0

for names in list1:

count\_a+=names.count("a")

print("Occurrence of a in given list is",count\_a)

OUTPUT:



RESULT:

Program to count the occurrences of ‘a’ within the list has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:6

COMPARE TWO LIST

Date: 05/12/2022

AIM: Enter 2 lists of integers. Check

(a) Whether lists are of same length.

(b) whether list sums to same value.

(c) whether any value occur in both.

ALGORITHM:

Step 1: Start.

Step 2: Input two strings.

Step 3: Get the length of two strings. Compare the lengths and print.

Step 4: Find the sum of elements in the list and compare and print.

Step 5: Find the similar value occurrence and print.

Step 6: Stop.

SOURCE CODE:

def length(flist,slist):

print("a.Length of list 1\t",len(flist))

print("\tLength of list 2\t",len(slist))

if len(flist)==len(slist):

print("\tBoth list have same size")

else:

print("Different length ")

def sumoflist(flist,slist):

s1=0

s2=0

for num in flist:

s1+=num

for num in slist:

s2+=num

if s1==s2:

print("b.Sum are same ",s1," ,",s2)

else:

print("b.Sum are different for both list ",s1," ,",s2)

def findele(flist,slist):

for num in flist:

if num in slist:

print("c.",num," found in both list\n")

flist=[]

slist=[]

len1=int(input("Enter the number of elements you want to add on list 1 "))

for i in range(0,len1):

print("Enter the element ",i+1)

inp=int(input())

flist.append(inp)

len2=int(input("Enter the number of elements you want to add on list 2 "))

for i in range(0,len2):

print("Enter the element ",i+1)

inp=int(input())

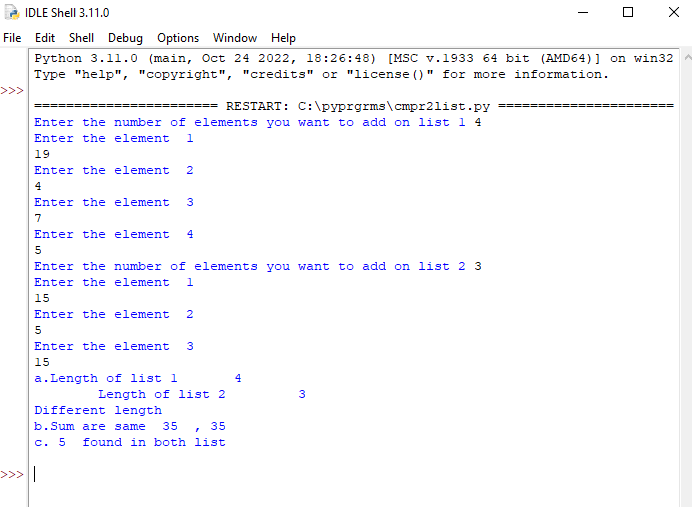
slist.append(inp)

length(flist,slist)

sumoflist(flist,slist)

findele(flist,slist)

OUTPUT:



RESULT:

Program to compare two list has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:7

CHARACTER REPLACE

Date: 05/12/2022

AIM: Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character. [eg: onion -> oni$n]

ALGORITHM:

Step 1: Start.

Step 2: Input the string.

Step 3: Replace the character with ‘$’ using replace().

Step 4: Print string.

Step 5: Stop.

SOURCE CODE:

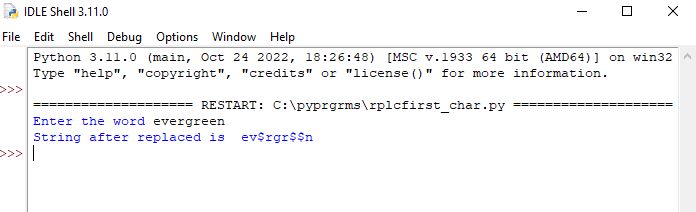
a=input("Enter the word ")

for i in range(1,len(a)):

b =a[0]+a[1:].replace(a[0], '$')

print("String after replaced is ",b)

OUTPUT:



RESULT:

Program to replace character has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:8

SWAP CHARACTERS

Date: 05/12/2022

AIM: Create a string from given string where first and last characters exchanged. [eg: python -> nythop]

ALGORITHM:

Step 1: Start.

Step 2: Input a string to a variable.

Step 3: Define a function,

Store last, first and middle character to a variable.

Print swapped string.

Step 4: Stop.

SOURCE CODE:

def swap(str1):

# storing the first character

start = str1[0]

# storing the last character

end = str1[-1]

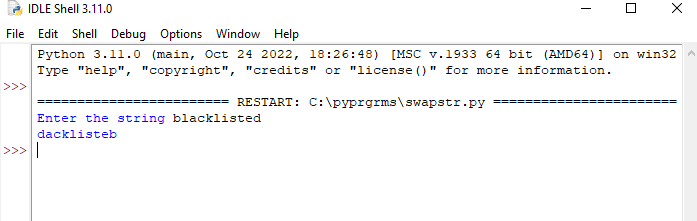
swapped\_string = end + str1[2:-1] + start

print(swapped\_string)

a=input("Enter the string ")

swap(a)

OUTPUT:



RESULT:

Program to swap first and last character of a string has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:9

AREA OF CIRLCE

Date: 05/12/2022

AIM: Accept the radius from user and find area of circle.

ALGORITHM:

Step 1: Start.

Step 2: Input radius.

Step 3: Compute 3.14\*r2.

Step 4: Print result.

Step 5: Stop.

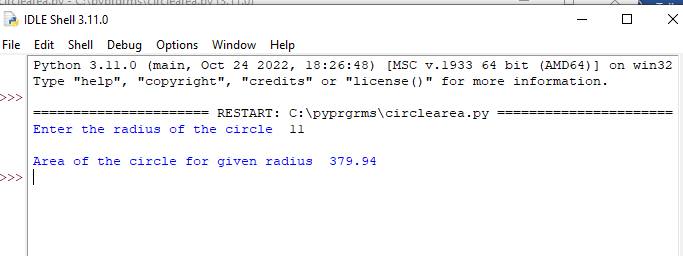
SOURCE CODE:

r=float(input("Enter the radius of the circle\t"))

area=3.14\*r\*\*2

print("\nArea of the circle for given radius ",area)

OUTPUT:



RESULT:

Program to find area of circle has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:10

BIGGEST OF 3 NUMBERS

Date: 05/12/2022

AIM: Find biggest of three numbers entered.

ALGORITHM:

Step 1: Start.

Step 2: Input 3 numbers.

Step 3: Check

If a>b and a>c:

Print(a)

If b>a and b>c:

Print(b)

Else:

Print(c)

Step 4: Stop.

SOURCE CODE:

print("Enter the 3 no:s")

a=int(input("\nEnter the 1st no: "))

b=int(input("\nEnter the 2nd no: "))

c=int(input("\nEnter the 3rd no: "))

if a>b and a>c:

print(a,"\t is biggest")

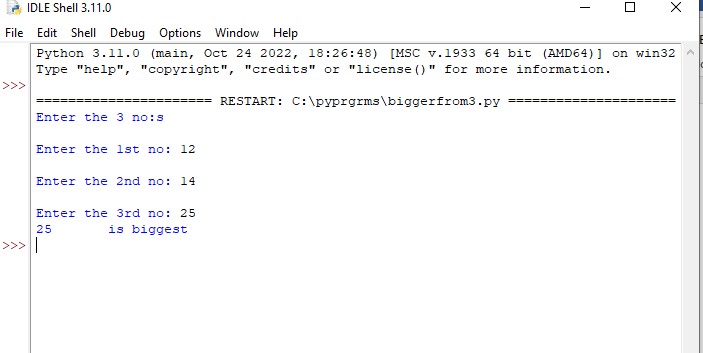
elif b>a and b>c:

print(b,"\t is biggest")

else:

print(c,"\t is biggest")

OUTPUT:



RESULT:

Program to find biggest of three numbers has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:11

GET FILE EXTENSION

Date: 05/12/2022

AIM: Accept a file name from user and print extension of that.

ALGORITHM:

Step 1: Start.

Step 2: Input a file name.

Step 3: Store the file name extension using split().

Step 4: Print extension.

Step 5: Stop.

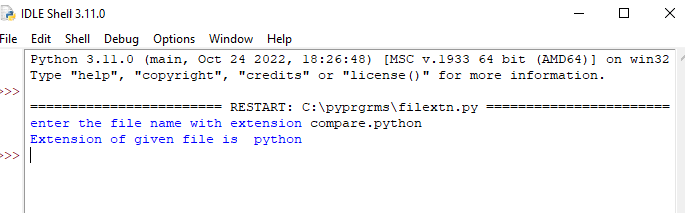
SOURCE CODE:

filename=input("enter the file name with extension")

extns=filename.split(".")

print("Extension of given file is ",extns[-1])

OUTPUT:



RESULT;

Program to get file extension has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:12

LIST OF COLOURS NAME

Date: 05/12/2022

AIM: Create a list of colors from comma-separated color names entered by user. Display first and last colors.

ALGORITHM:

Step 1: Start.

Step 2: Input a list containing name of colors.

Step 3: Store first and last element of list value and print.

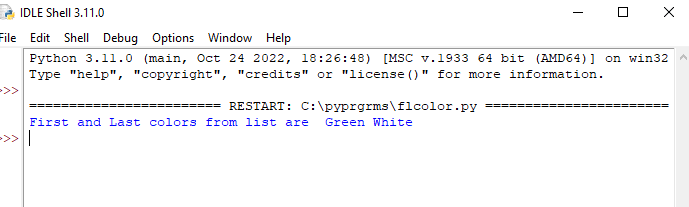
Step 4: Stop.

SOURCE CODE:

clr=["Green","Red","White"]

print("First and Last colors from list are ",clr[0],clr[-1])

OUTPUT:



RESULT:

Program to display first and last colors has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:13

COMPUTE EXPRESSION.

Date: 05/12/2022

AIM: Accept an integer n and compute n+nn+nnn.

ALGORITHM:

Step 1: Start.

Step 2: Input value of n, initialize a variable s =0 to store sum of the expression.

Step 3: Compute s=s+n\*\*i with range(1,4).

Step 4: Print result.

Step 5: Stop.

SOURCE CODE:

n=int(input("\nEnter the number to compute"))

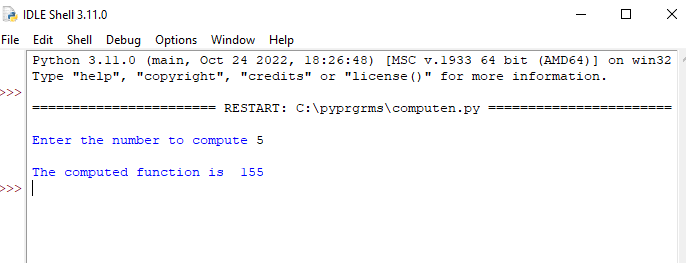
s=0

for i in range(1,4):

s=s+n\*\*i

print("\nThe computed function is ",s)

OUTPUT:



RESULT:

Program to compute a function has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:14

DISPLAY DIFFERENCE OF TWO LIST

Date: 05/12/2022

AIM: Print out all colors from color-list1 not contained in color-list2.

ALGORITHM:

Step 1: Start.

Step 2: Input two list with elements as set().

Step 3: Using difference() .

Print list of elements.

Step 4: Stop.

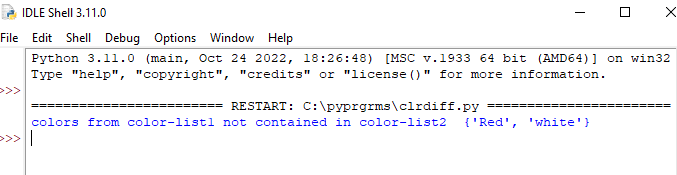
SOURCE CODE:

clr1=set(["Green","Red","white"])

clr2=set(["Pink","Aqua","Green"])

print("colors from color-list1 not contained in color-list2 ",clr1.difference(clr2))

OUTPUT:



RESULT:

Program to display difference of list has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:15

SWAP TWO STRINGS

Date: 05/12/2022

AIM: Create a single string separated with space from two strings by swapping the character at position 1.

ALGORITHM:

Step 1: Start.

Step 2: Define function to swap.

Step 3: Store each string to new variable.

Step 4: Print resulted string.

Step 5: Input two string.

Step 6: Call the function.

Step 7: Stop.

SOURCE CODE:

def charswap(a, b):

new\_a = b[:1] + a[1:]

new\_b = a[:1] + b[1:]

return new\_a + ' ' + new\_b

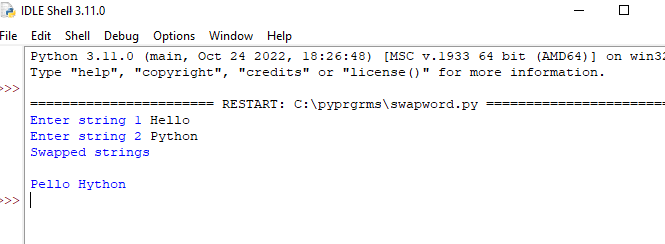
a=input("Enter string 1 ")

b=input("Enter string 2 ")

print("Swapped strings\n")

print(charswap(a,b))

OUTPUT:



RESULT:

Program to swap string character has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:16

SORT DICTIONARY

Date: 08/12/2022

AIM: Sort dictionary in ascending and descending order.

ALGORITHM:

Step 1: Start.

Step 2: Input a dictionary.

Step 3: Convert a given dictionary to list, l.

Step 4: To sort in ascending order.

Call l.sort()

For descending order.

Call l.sort(reverse=True)

Step 5: dict=dict(1).

Step 6: Stop.

SOURCE CODE:

d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

print('Original dictionary : ',d)

l=list(d.items())

#convert the given dict. into list

l.sort() #sort the list

print(“Ascending order is ”,l)

l=list(d.items())

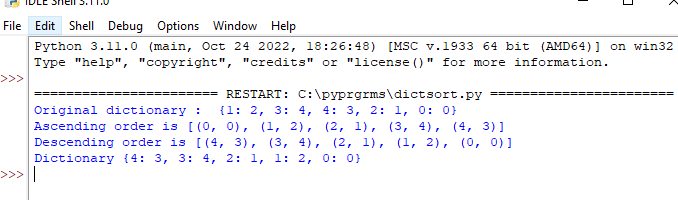
l.sort(reverse=True) #sort in reverse order

print(“Descending order is ”,l)

dict=dict(l) # convert the list in dictionary

print("Dictionary ",dict)

OUTPUT:



RESULT:

Program to sort dictionary has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:17

MERGE DICTIONARY

Date: 08/12/2022

AIM: Merge two dictionaries.

ALGORITHM:

Step 1: Start.

Step 2: Input 2 dictionaries d and c.

Step 3: Call update()

d.update(c)

Step 4: Print(d).

Step 5: Stop.

SOURCE CODE:

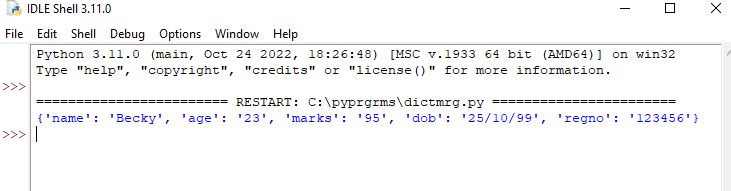
d={"name":"Becky","age":"23","marks":"95"}

c={"dob":"25/10/99","regno":"123456"}

d.update(c)

print(d)

OUTPUT:



RESULT:

Program to merge two dictionaries has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:18

GCD OF 2 NUMBERS

Date: 08/12/2022

AIM: Find GCD of two numbers

ALGORITHM:

Step 1: Start.

Step 2: Define function.

Step 3: Check if x>y set

s=y

else set

s=x

Step 4: Check until 1,s+1

if ((x%i==0) and (y%i==0))

Print gcd

Step 5: Call function.

Step 6: Stop.

SOURCE CODE:

def compute\_gcd(x,y):

if x>y:

s=y

else:

s=x

for i in range(1,s+1):

if ((x%i==0) and (y%i==0)) :

gcd=i

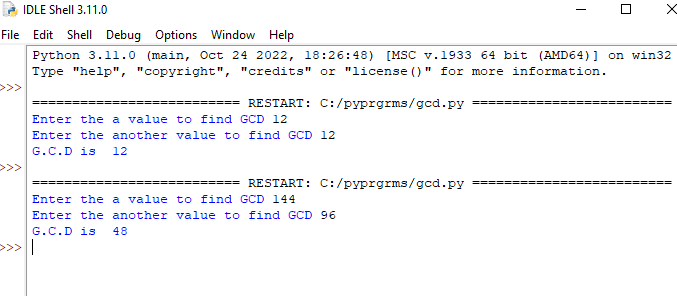
print("G.C.D is ",gcd)

a=int(input("Enter the a value to find GCD "))

b=int(input("Enter the another value to find GCD "))

compute\_gcd(a,b)

OUTPUT:



RESULT:

Program to find GCD has been executed successfully and output is verified.

LABCYCLE 1

EXPERIMENT NO:19

LIST OF EVEN INTEGERS

Date: 08/12/2022

AIM: From a list of integers, create a list removing even numbers.

ALGORITHM:

Step 1: Start.

Step 2: Input limit of list,n.

Step 3: Append element to the list until 0,n.

Step 4: Check, for i l.

i%2!=0 then,

Append to list even.

Step 5: Print even.

Step 6: Stop.

SOURCE CODE:

l=[]

even=[]

n=int(input("Enter the limit of list "))

print("Enter the integers into the list ")

for i in range(0,n):

print("Enter the element no: -{}".format(i+1))

elm=int(input())

l.append(elm)

print("List of integers are ",l)

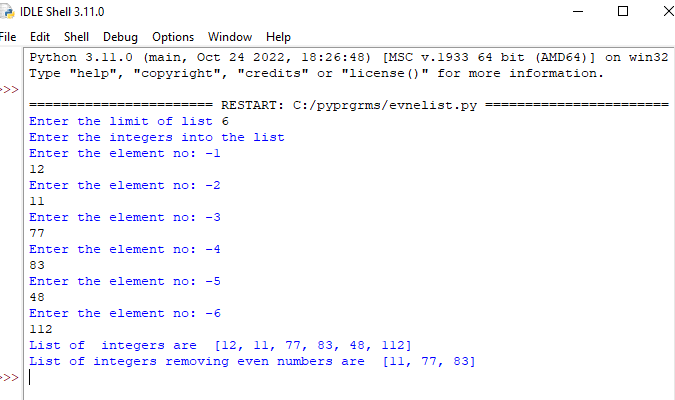
for i in l:

if i%2!=0:

even.append(i)

print("List of integers removing even numbers are ",even)

OUTPUT



RESULT:

Program to display list of integers removing even numbers has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:1

FACTORIAL

Date: 12/12/2022

AIM: Program to find the factorial of a number.

ALGORITHM:

SORCE CODE:

def fact(num):

f=1

if num==0:

print("Factorial is : ",1)

elif num<0:

print("Can't find the factorial ")

else:

for i in range(1,num+1):

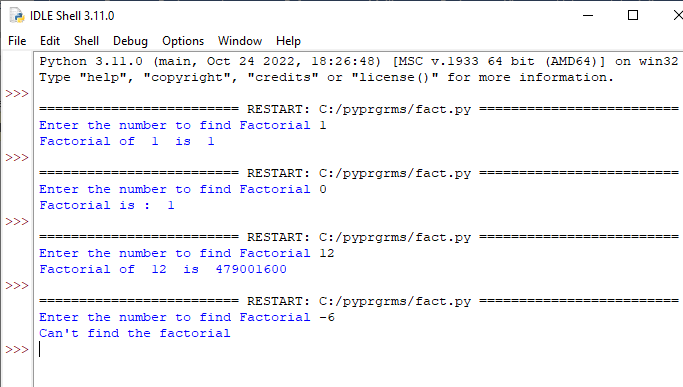
f=f\*i

print("Factorial of ", num," is ",f)

num=int(input("Enter the number to find Factorial "))

fact(num)

OUTPUT:



RESULT:

Program to find factorial of a number has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:2

FIBONACCI

Date: 12/12/2022

AIM: Generate Fibonacci series of N terms.

ALGORITHM:

SORCE CODE:

a=int(input("enter the 1st no in the series "))

b=int(input("enter the 2nd no in the series "))

n=int(input("enter the no: of terms needed "))

print("Fibonacci series ")

print(a,b, end=" ")

while n-2:

c=a+b

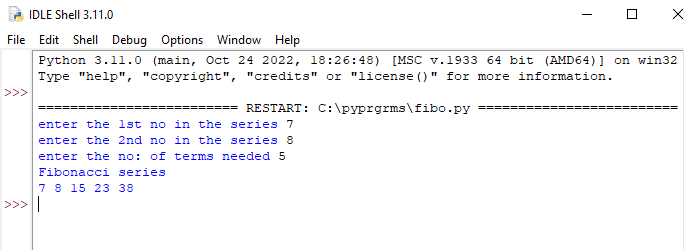
a=b

b=c

print(c, end=" ")

n=n-1

OUTPUT:



RESULT:

Program to generate Fibonacci has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:3

SUM OF ITEMS IN A LIST

Date: 12/12/2022

AIM: Find the sum of all items in a list.

ALGORITHM:

SORCE CODE:

list1=[]

len1=int(input("Enter the number of elements you want to add on list 1 "))

for i in range(0,len1):

print("Enter the element ",i+1)

inp=int(input())

list1.append(inp)

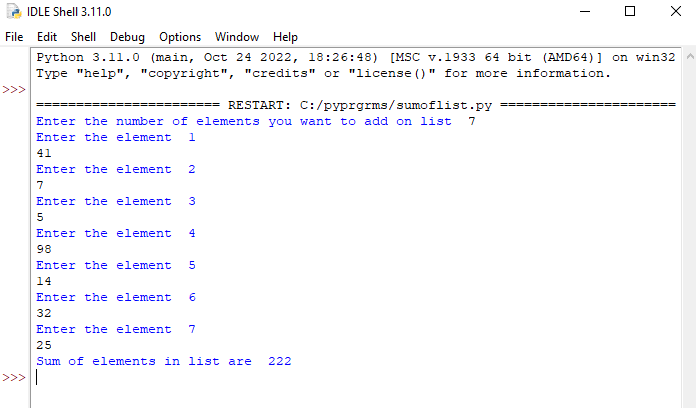
s=0

for i in list1:

s=s+i

print("Sum of elements in list are ",s)

OUTPUT:



RESULT:

Program to find sum of items in a list has been executed successfully and output is verified.