DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

20MCA131 - PROGRAMMING LAB

PRACTICAL RECORD BOOK

First Semester MCA

2020-2021

Submitted by:

NAME : ARJUN V PANKAJAKSHAN

ROLL NO : MCA138

DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

Certificate

This is a bonafide record of the work done by ARJUN V PANKAJAKSHAN in the First Semester in Programming Lab Course(20MCA131) towards the partial fulfillment of the degree of Master of Computer Applications during the academic year 2020-2021.

Staff Member in-charge

Examiner

……..

……..

INDEX

PROGRAM 1 : FINDING LEAP YEARS

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

ALGORITHM :

Step 1: Read current year and ending year as inputs.

Step 2: If current year < ending year.

Step 3: Initialise counter variable i=current_year.

Step 4: If i<(end_year+1) go to step 5.Else go to step 7

Step 5: If i%4=0 ,then print i, else go to step 6.

Step 6: Increment counter variable and go to step 4.

Step 7: Stop.

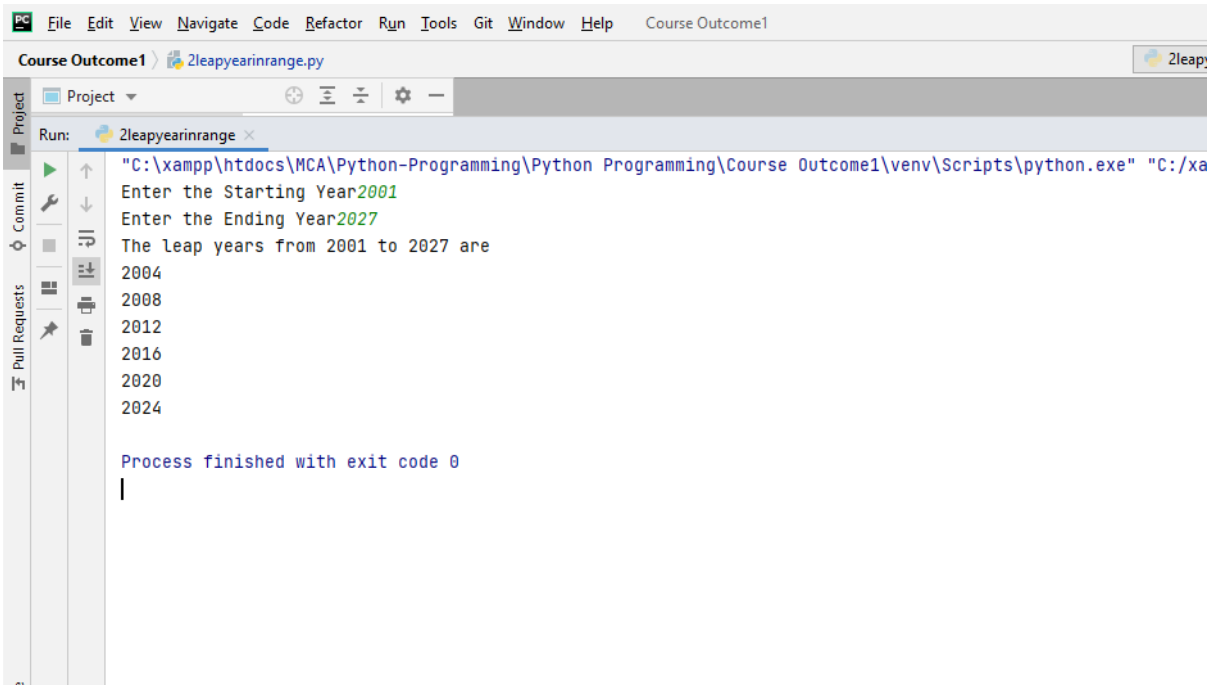
PROGRAM CODE :

2leapyearinrange	start=int(input("Enter the Starting Year"))
	end=int(input("Enter the Ending Year"))
	print("The leap years from "+str(start)+" to "+str(end)+" are ")
	if start<end:
	  for i in range(start,end+1):
	  if i%4==0:
	  print(i)

RESULT :

The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help Course Outcome1
Course Outcome1 2leapyearinrange.py
Project
Run: 2leapyearinrange
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\Course Outcome1\venv\Scripts\python.exe" "C:/xa
Enter the Starting Year2001
Enter the Ending Year2027
The leap years from 2001 to 2027 are
2004
2008
2012
2016
2020
2024
Process finished with exit code 0

PROGRAM 2 : LIST COMPREHENSIONS

(A) LIST OF POSITIVE NUMBERS

AIM : Generate positive list of numbers from a given list of integers

ALGORITHM :

Step 1: Read items into list1.

Step 2: Check for items that are greater than zero and store in list2.

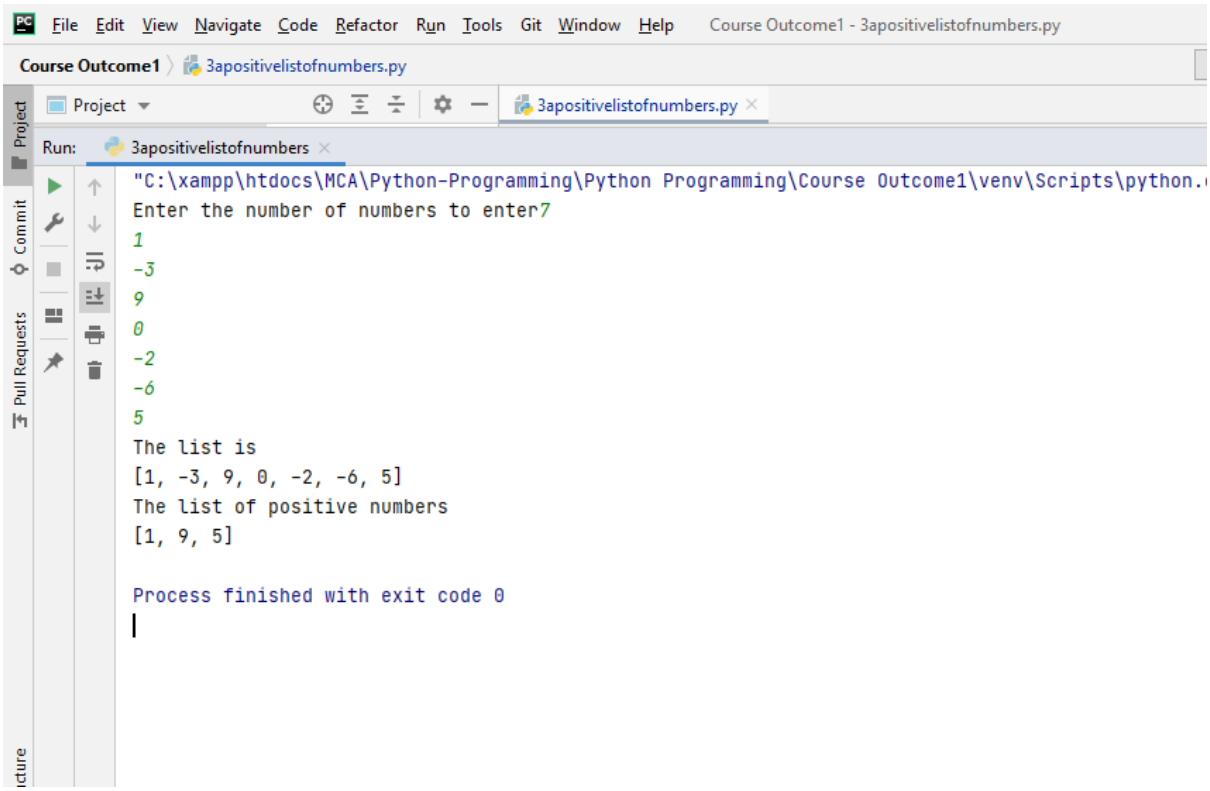
Step 3: Print list2.

PROGRAM CODE :

3apositivelistofnumbers.py	newlist=[]
	list=[]
	n = int(input("Enter the number of numbers to enter"))
	for i in range(n):
	  list.append(int(input()))
	print("The list is")
	print(list)
	print("The list of positive numbers")
	for i in range(n):
	  if list[i]>0:
	  newlist.append(list[i])
	print(newlist)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
Course Outcome1 - 3apositivelistofnumbers.py
Course Outcome1
3apositivelistofnumbers.py
Project
3apositivelistofnumbers.py
Run:
3apositivelistofnumbers
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\Course Outcome1\venv\Scripts\python.
Enter the number of numbers to enter7
1
-3
9
0
-2
-6
5
The list is
[1, -3, 9, 0, -2, -6, 5]
The list of positive numbers
[1, 9, 5]
Process finished with exit code 0

(B) SQUARE OF N NUMBERS

AIM : Square of N numbers

ALGORITHM :

Step 1: Print squares of numbers from 0 to 10 and store in list.

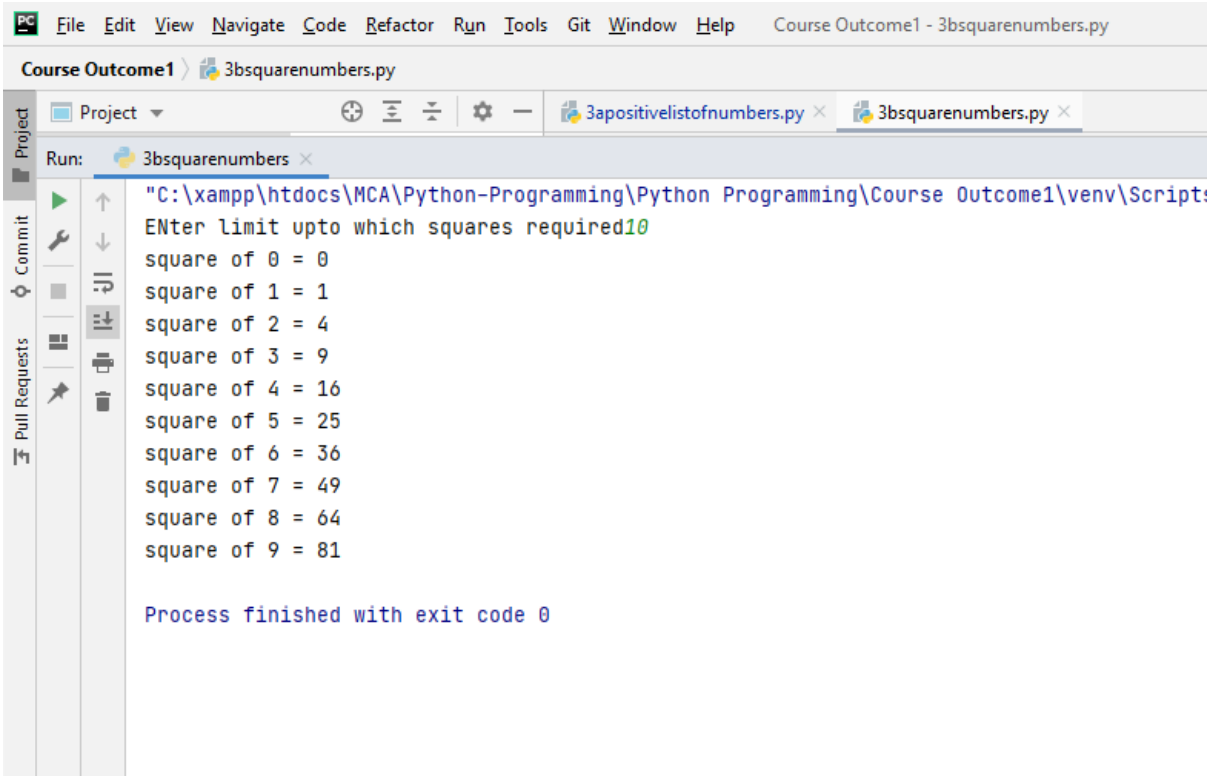
Step 2: Print list.

PROGRAM CODE :

3bsquarenumbers.py	n=int(input("Enter limit upto which squares required"))
	for i in range(n):
	  square=i**2
	  print("square of "+str(i)+" = "+str(square))

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
Course Outcome1 - 3bsquarenumbers.py
Course Outcome1
3bsquarenumbers.py
Project
3apositivelistofnumbers.py
3bsquarenumbers.py
Run:
3bsquarenumbers
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\Course Outcome1\venv\Scripts
ENter limit upto which squares required10
square of 0 = 0
square of 1 = 1
square of 2 = 4
square of 3 = 9
square of 4 = 16
square of 5 = 25
square of 6 = 36
square of 7 = 49
square of 8 = 64
square of 9 = 81
Process finished with exit code 0
Project
Commit
Pull Requests

(C) LIST OF VOWELS

AIM : Form a list of vowels selected from a given word.

ALGORITHM :

Step 1: Input a string.

Step 2: Assign an array vowels.

Step 3: Check for vowels present in the string and store in list.

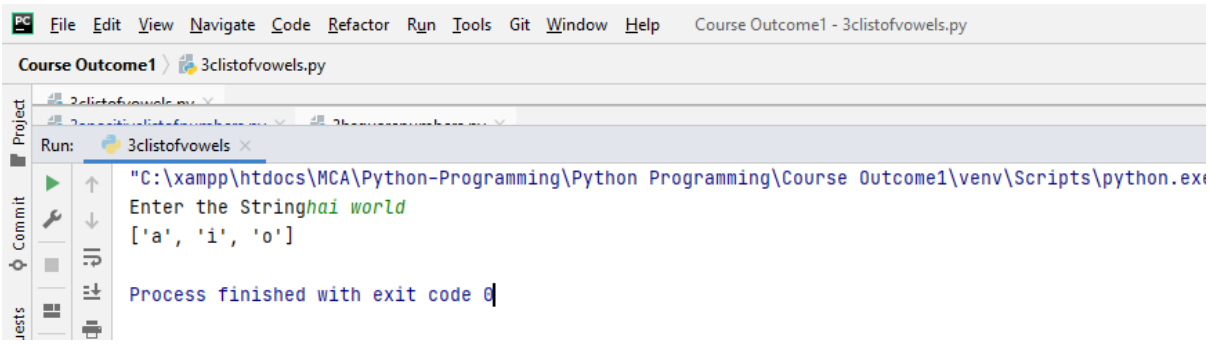
Step 4: Print list.

PROGRAM CODE :

3clistofvowels.py	vow=[]
	string=str(input("Enter the String"))
	for i in range(len(string)):
	  if (string[i]=="a" or string[i]=="e" or string[i]=="i" or string[i]=="o" or string[i]=="u"):
	  vow.append(string[i])
	print(vow)

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :



(D) ORDINAL VALUE OF ELEMENTS

AIM : List ordinal value of each element of a word.

ALGORITHM :

Step 1: Input a string.

Step 2: Assign ordinal value for each element and store in list.

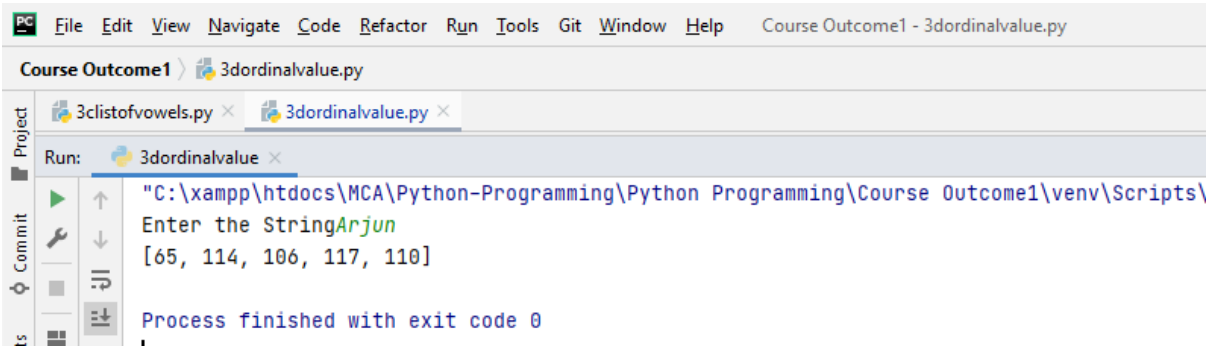
Step 3: Print list.

PROGRAM CODE :

3dordinalvalue.py	list=[] word=input("Enter the String") for i in range(len(word)): list.append(ord(word[i])) print(list)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 3 : OCCURENCES OF A WORD

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

ALGORITHM :

Step 1: Input a string.

Step 2: Split the words.

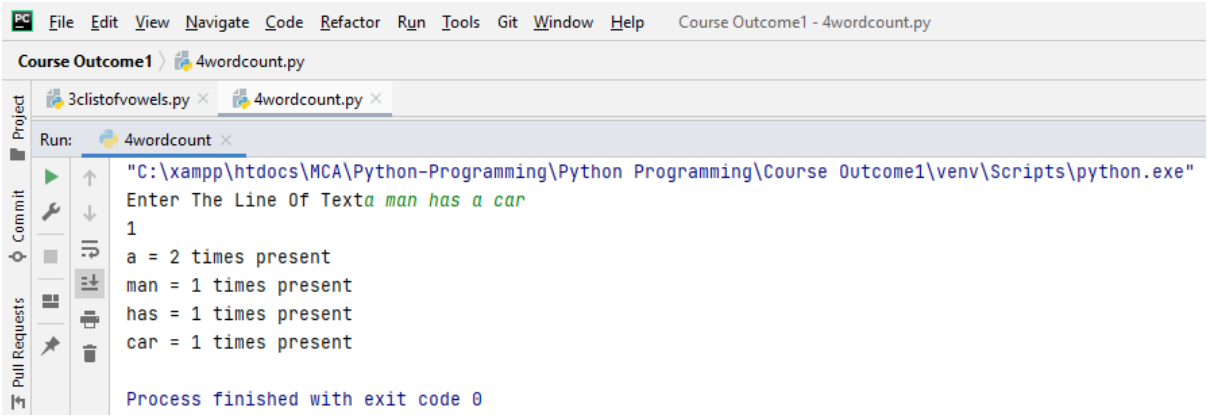
Step 3: Count the number of words and print.

PROGRAM CODE :

4wordcount.py	res=[]
	lineoftext=input("Enter The Line Of Text")
	list=lineoftext.split(" ")
	for i in range(len(list)):
	  list.count(list[i])
	print(list.count(list[i]))
	for i in list:
	  if i not in res:
	  res.append(i)
	for i in range(len(res)):
	  print("%s = %s times present" % (res[i], list.count(res[i])))list :
	"+str(len(list)))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
Course Outcome1 - 4wordcount.py
Course Outcome1
4wordcount.py
3clistofvowels.py
4wordcount.py
Run:
4wordcount
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\Course Outcome1\venv\Scripts\python.exe"
Enter The Line Of Texta man has a car
1
a = 2 times present
man = 1 times present
has = 1 times present
car = 1 times present
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 4 : STORING ANOTHER WORD FOR INCORRECT VALUE

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

ALGORITHM :

Step 1: Input a number.

Step 2: Check if number >10,Print “over” .Else print number.

PROGRAM CODE :

5numbergreaterthan100.py	a=[]
	list=[]
	n=int(input("Enter the number of numbers to enter"))
	for i in range(n):
	  x=int(input())
	  if x<=100:
	  list.append(x)
	  else:
	  list.append("over")
	print(list)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
Course Outcome1 - 5numbergreaterthan100.py
Course Outcome1
5numbergreaterthan100.py
3clistofvowels.py
4wordcount.py
5numbergreaterthan100.py
Run:
5numbergreaterthan100
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\Course Outcome1\venv\Scripts\py
Enter the number of numbers to enter4
98
110
3
172
[98, 'over', 3, 'over']
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 5 : NUMBER OF OCCURENCES OF LETTER ‘a’ IN A WORD

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

ALGORITHM :

Step 1: Input number of names.

Step 2: Input names.

Step 3: Check for count of ‘a’ in every words.

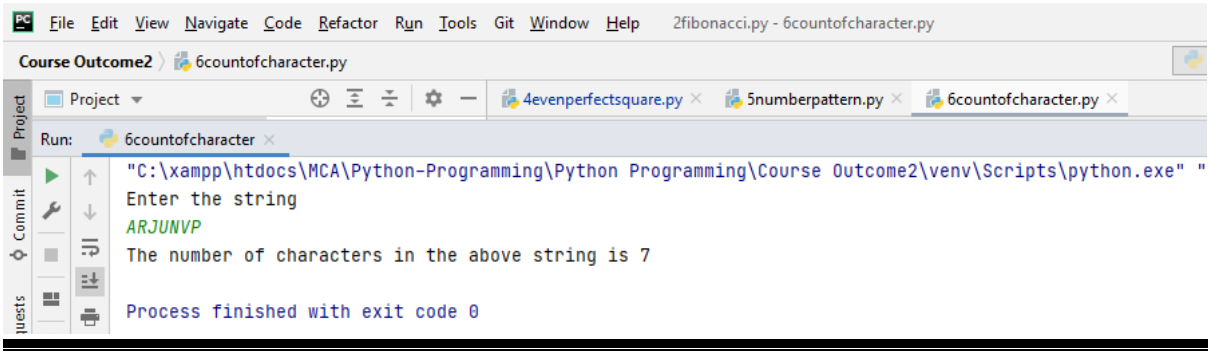
Step 4: Print the count.

PROGRAM CODE :

6occuranceofainlist.py	names = input("Enter the name seperated by a comma:")
	x = names.split(",")
	c = 0
	print("x =",x)
	for i in x:
	  for n in i:
	  if n == 'a':
	  c = c + 1
	print("Counts of 'a':",c)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 6 : OPERATIONS IN LIST

AIM : Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both

ALGORITHM :

Step 1: Input lists list1 and list2.

Step 2: Check if length of list1 and list2 are same or not.

Step 3: Check if sum of lists list1 and list2 are same or not.

Step 4: Find the common values in list1 and list2.

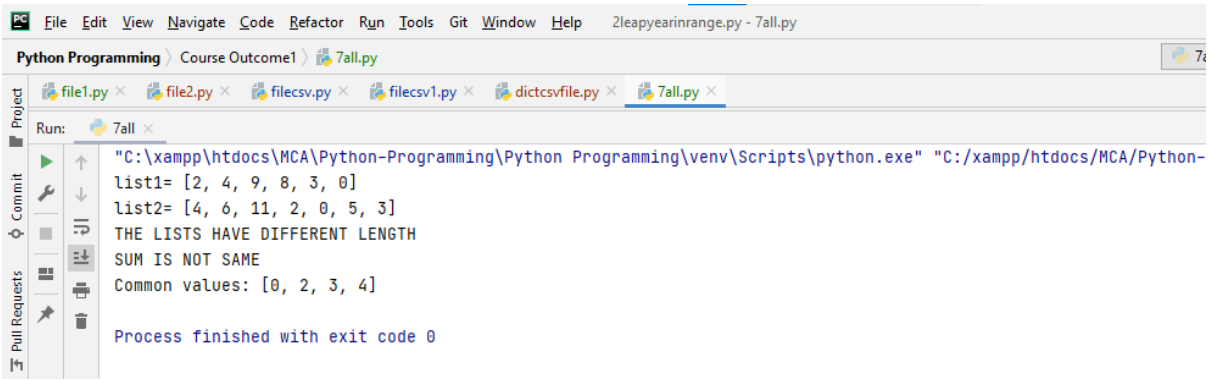
PROGRAM CODE :

7all.py

```
list1 = [2,4,9,8,3,0]
list2 = [4,6,11,2,0,5,3]
print("list1=",list1)
print("list2=",list2)
a = len(list1)
b = len(list2)

if a == b:
   print("THE LISTS HAVE SAME LENGTH")
else:
   print("THE LISTS HAVE DIFFERENT LENGTH")

s1 = sum(list1)
s2 = sum(list2)
if s1 == s2:
   print("THE TWO LISTS HAVE THE SAME SUM ")
else:
      print("SUM IS NOT SAME")

list1 = set(list1)
list2 = set(list2)
i = list1.intersection(list2)
i = list(i)
print("Common values:",i)
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 7 : REPLACING CHARACTER WITH $

AIM : Get a string from an input string where all occurrences of first character replaced with '$', except first character.

ALGORITHM :

Step 1: Input a word.

Step 2: Extract first character.

Step 3: Check for occurrence of the same character in the word, if found replace with $.

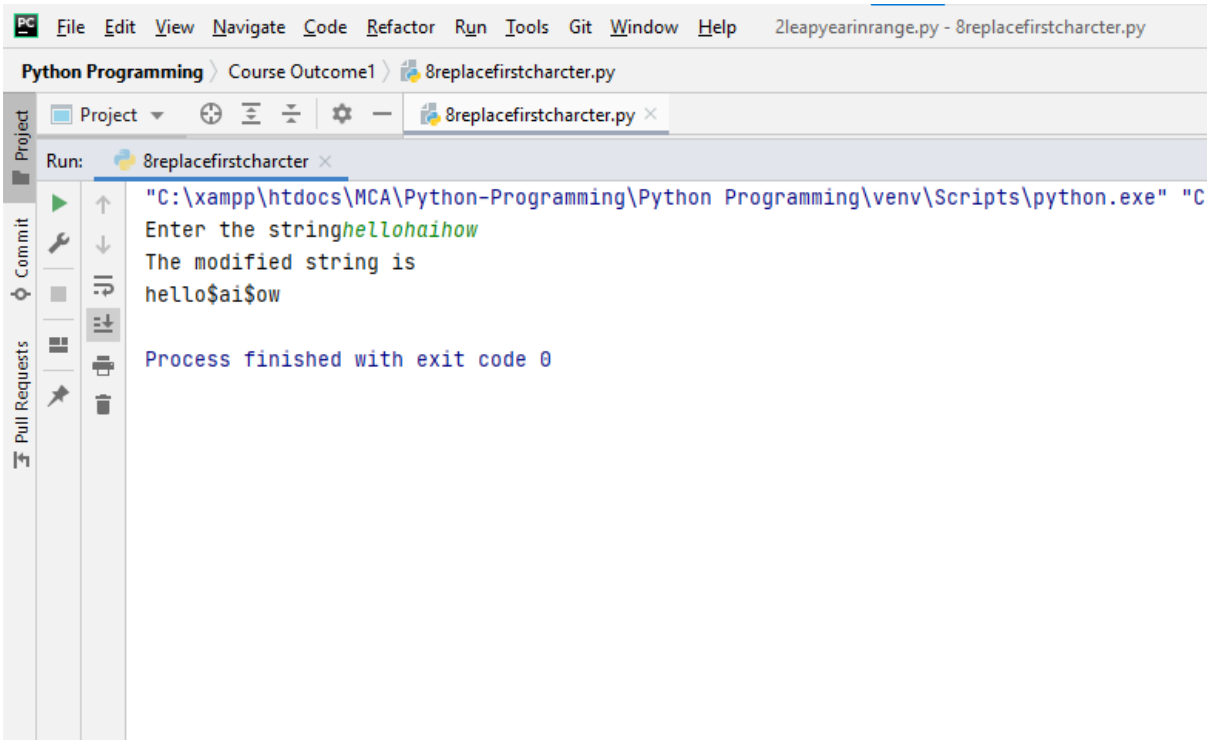
Step 4: Print word.

PROGRAM CODE :

8replacefirstcharcter.py	string=input("Enter the string") first=string[0] string=string.replace(first,"$") string=first+string[1:] print("The modified string is\n"+string)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 8replacefirstcharcter.py
Python Programming
Course Outcome1
8replacefirstcharcter.py
Project
8replacefirstcharcter.py
Run:
8replacefirstcharcter
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C
Enter the stringhellohaihow
The modified string is
helloaiow
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 8 : EXCHANGING FIRST AND LAST CHARACTER IN A STRING

AIM : Create a string from given string where first and last characters exchanged.

ALGORITHM :

Step 1: Input a string.

Step 2: Swap the first and last characters.

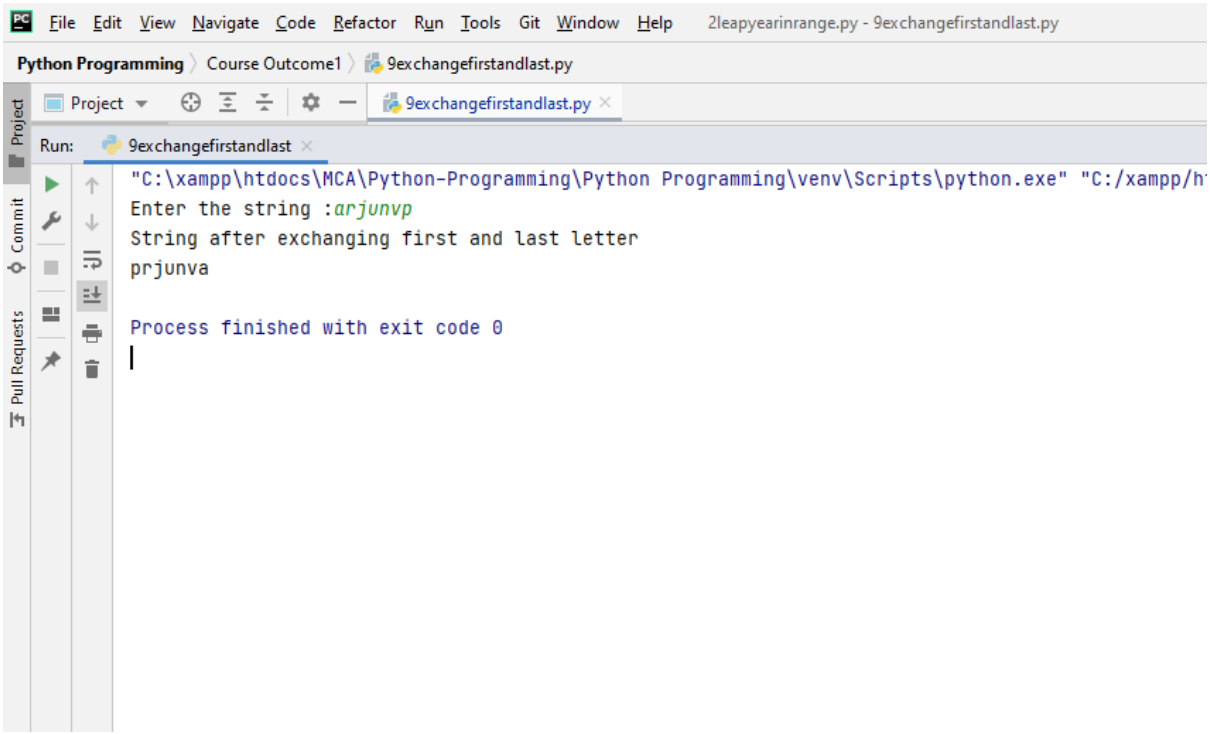
Step 3: Print String.

PROGRAM CODE :

9exchangefirstandlast.py	string=input("Enter the string :") string=string[-1]+string[1:(len(string)-1)]+string[0] print("String after exchanging first and last letter\n"+string)

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 9exchangefirstandlast.py
Python Programming
Course Outcome1
9exchangefirstandlast.py
Project
9exchangefirstandlast.py
Run:
9exchangefirstandlast
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/h
Enter the string :arjunvp
String after exchanging first and last letter
prjunva
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 9 : AREA OF CIRCLE

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

ALGORITHM :

Step 1: Input radius.

Step 2: Compute area of circle.

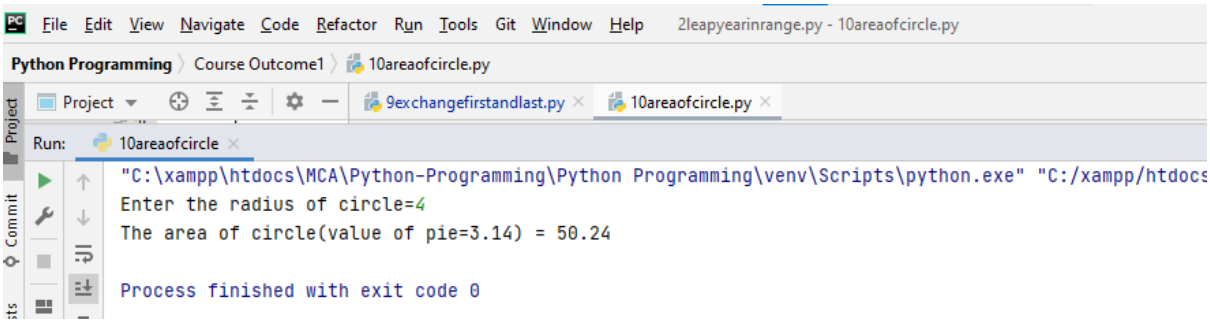
Step 3: Print area.

PROGRAM CODE :

10areaofcircle.py	radius=int(input("Enter the radius of circle=")) area=3.14*radius**2 print("The area of circle(value of pie=3.14) = "+str(area))

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :



PROGRAM 10: BIGGEST OF 3 NUMBERS

AIM : Find biggest of 3 numbers entered.

ALGORITHM :

Step 1: Take first three numbers as input .

Step 2: Store it in an array.

Step 3: Sort the array components in descending order.

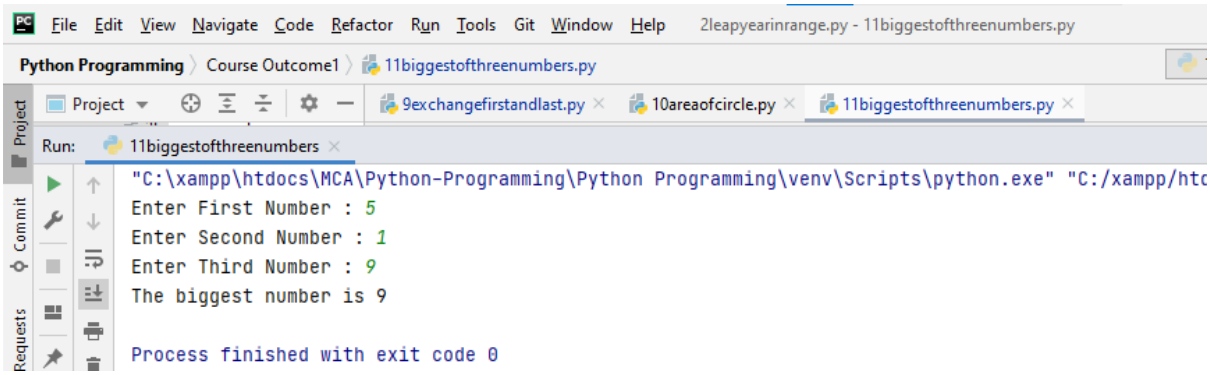
Step 4: Print the array[0] as biggest element.

PROGRAM CODE :

11biggestofthreenumbers.py	num1= int(input("Enter First Number : ")) num2= int(input("Enter Second Number : ")) num3= int(input("Enter Third Number : ")) array = [num1,num2,num3] array.sort(reverse=True) print("The biggest number is",array[0])

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 11: EXTENSION OF A FILE

AIM : Find extension of the file entered by the user.

ALGORITHM :

Step 1: Take any filename as input.

Step 2: Using rfind function find the extension name after the ‘ .’ symbol

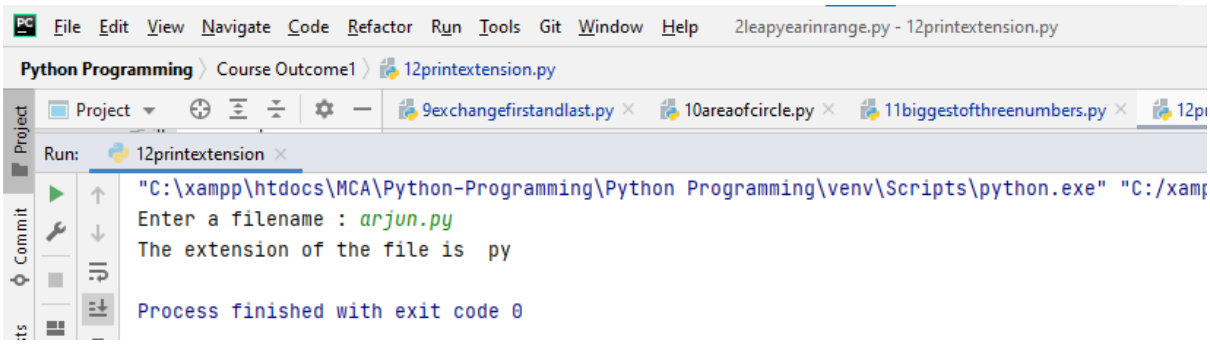
Step 3: Print the extension[1:]

PROGRAM CODE :

12printextension.py	filename = input("Enter a filename : ") position = filename.rfind(".") extension = filename[position:] print("The extension of the file is ",extension[1:])

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 12: COLOR SEPARATION

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

ALGORITHM :

Step 1: Enter the colors separated by comma.

Step 2: Split the colours using split() function and Store the colors in list.

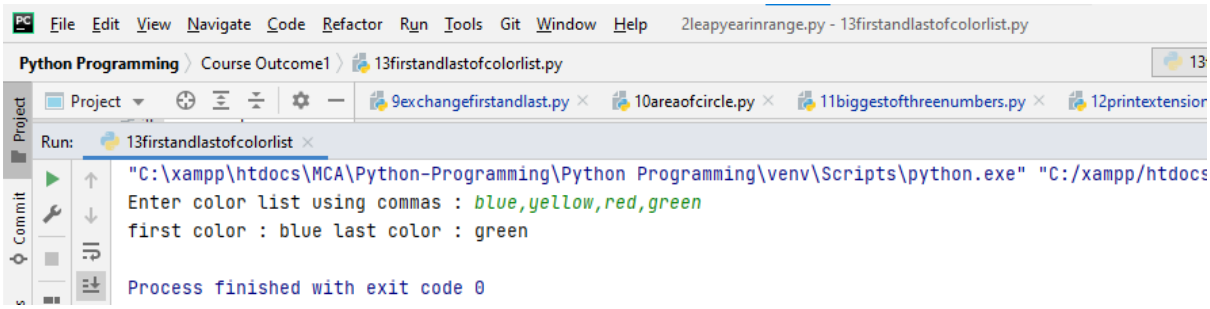
Step 4: Print first and last color using list index .

PROGRAM CODE :

13firstandlastofcolorlist.py	colors = input("Enter color list using commas : ") list = colors.split(",") print("first color : "+list[0]+" last color : "+list[len(list)-1])

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 13: INTEGER OPERATION

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

ALGORITHM :

Step 1: Read a number.

Step 2: Calculate value=n+n*n+n*n*n

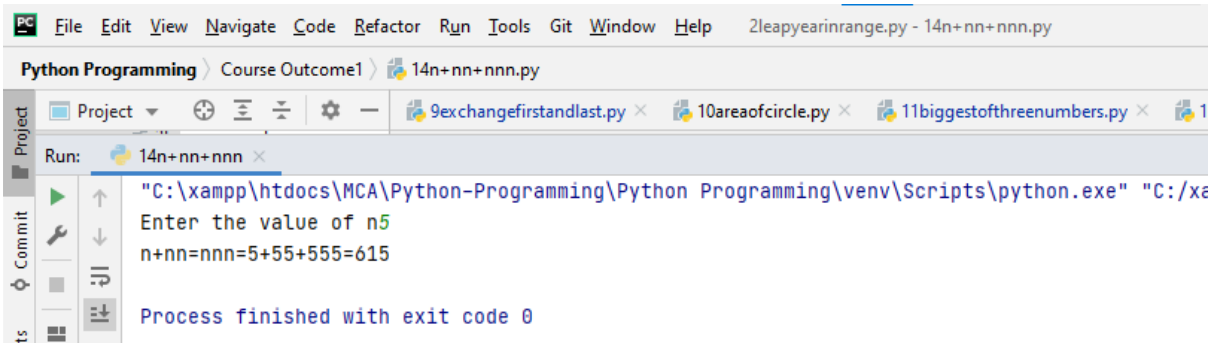
Step 3: Print value.

PROGRAM CODE :

14n+nn+nnn.py	n=int(input("Enter the value of n")) value=n+((n*10)+n)+(((n*100)+n*10)+n) print("n+nn=nnn=%s+%s+%s=%s"%(n,((n*10)+n),(((n*100)+n*10)+n),value))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 14: COLOR LISTS

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

ALGORITHM :

Step 1: Input the colors in separate lists.

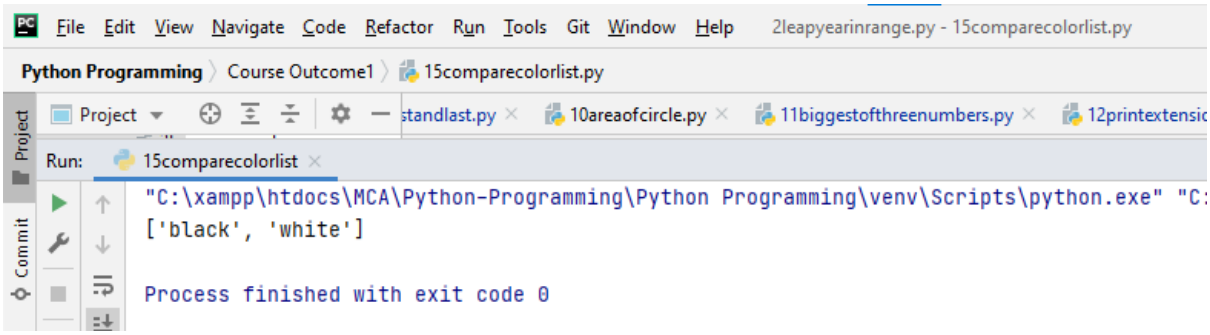
Step 2: Print colors that are present in list1 and not in list2.

PROGRAM CODE :

15comparecolorlist.py	list1=["black","red","white","yellow"] list2=["yellow","red","blue"] print([item for item in list1 if item not in list2])

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :



PROGRAM 15: CHARACTER SWAP

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

ALGORITHM :

Step 1: Input a string with 2 words separated by comma.

Step 2: Split the string by split function.

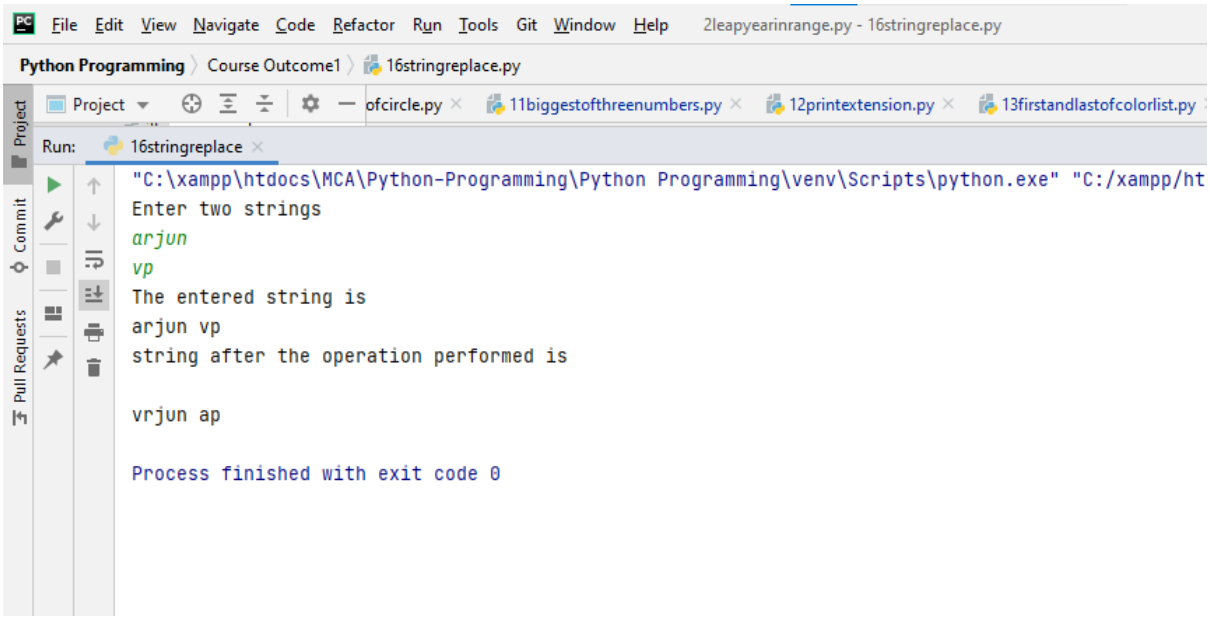
Step 3: Swap the first character of the first word with the first character of the second word.

PROGRAM CODE :

16stringreplace.py	word = input("Type a string with 2 words seperated by comma : ") word_list = word.split(" ") first_letter_1 = word_list[0][0] first_letter_2 = word_list[1][0] print(first_letter_2+word_list[0][1:]+" "+first_letter_1+word_list[1][1:])

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 16stringreplace.py
Python Programming
Course Outcome1
16stringreplace.py
Project
ofcircle.py
11biggestofthreenumbers.py
12printextension.py
13firstandlastofcolorlist.py
Run:
16stringreplace
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/ht
Enter two strings
arjun
vp
The entered string is
arjun vp
string after the operation performed is
vrjun ap
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 16: DICTIONARY SORTING

AIM : Sort dictionary in ascending and descending order.

ALGORITHM :

Step 1: Create a dictionary with key as letter and value for the corresponding letter.

Step 2: Sort the dictionary in ascending and descending order using sorted() function.

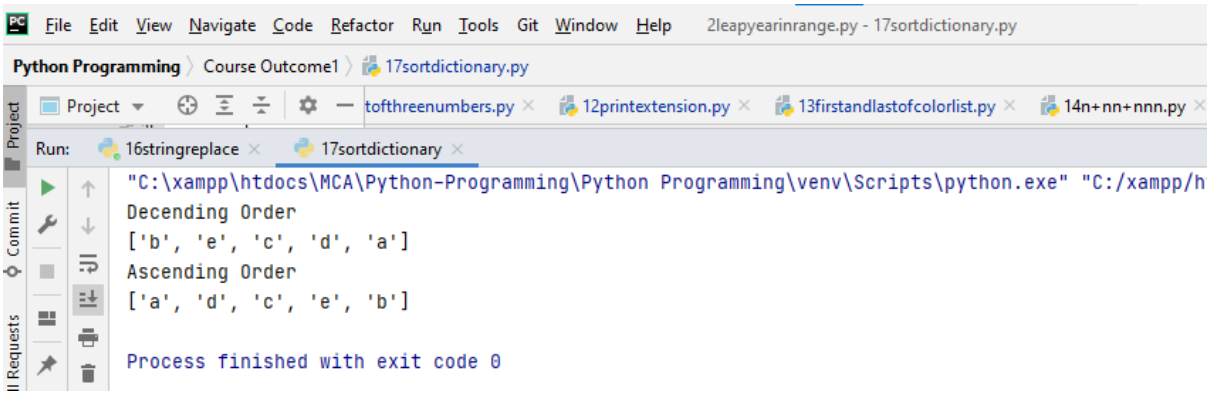
Step 3: Print sorted dictionary.

PROGRAM CODE :

17sortdictionary.py	a1 = {'a':2, 'b':31, 'd':4, 'c':22, 'e':30} a1_sorted_keys = sorted(a1, key=a1.get, reverse=True) a1_sorted_keys_2 = sorted(a1, key=a1.get) print("Decending Order") print(a1_sorted_keys) print("Ascending Order") print(a1_sorted_keys_2)

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :



PROGRAM 17: MERGING DICTIONARIES

AIM : Merge two dictionaries.

ALGORITHM :

Step 1: Declare two dictionaries with key and values.

Step 2: Appending second dictionary to first using the update() function.

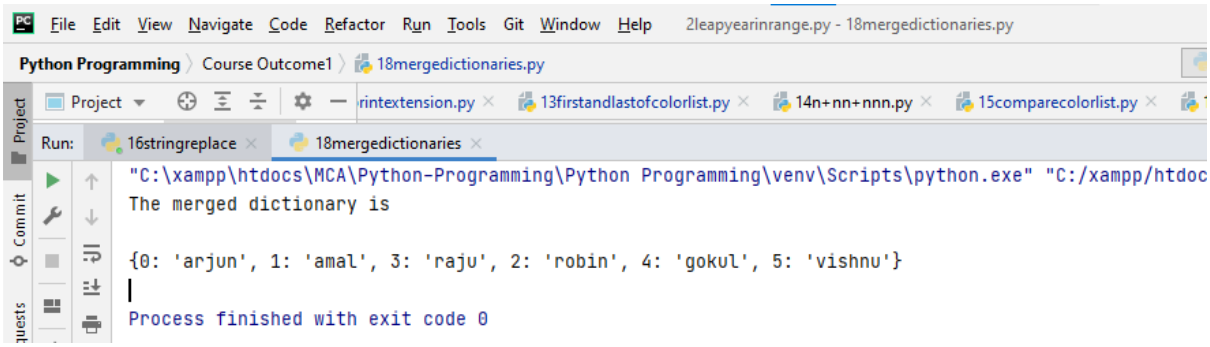
Step 3: Print the merged dictionary.

PROGRAM CODE :

18mergedictionaries.py	dict1={0:"arjun",1:"amal",3:"raju"} dict2={2:"robin",4:"gokul",5:"vishnu"} dict1.update(dict2) print("The merged dictionary is\n") print(dict1)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 18:GCD

AIM : Find gcd of 2 numbers.

ALGORITHM :

Step 1: Import math library.

Step 2: Input two numbers.

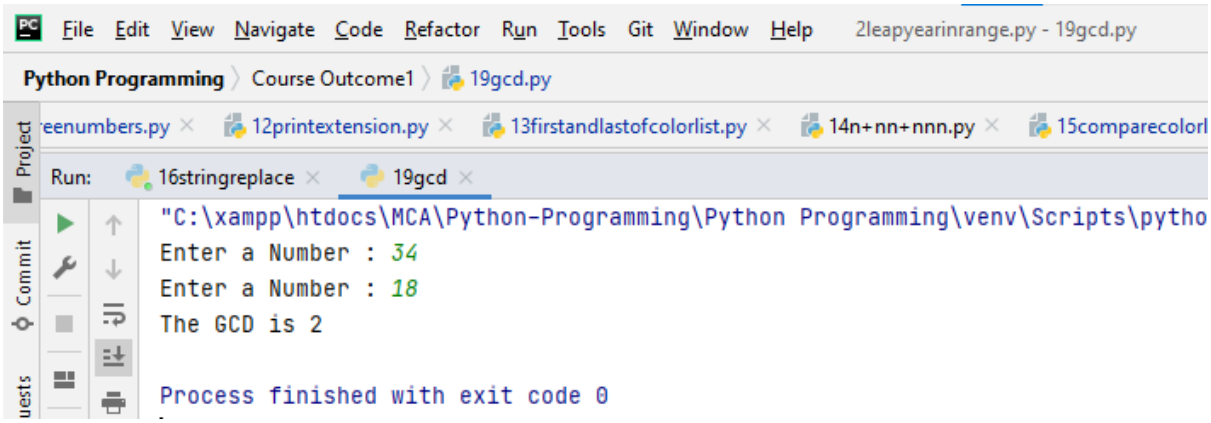
Step 3: Perform gcd operation.

PROGRAM CODE :

19gcd.py	import math a = int(input("Enter a Number : ")) b = int(input("Enter a Number : ")) print("The GCD is",math.gcd(a,b))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 19: REMOVAL OF EVEN NUMBERS

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

ALGORITHM :

Step 1: Declare a list with values and declare another empty list1.

Step 2: if elements in list is odd then ,it appends to list1.

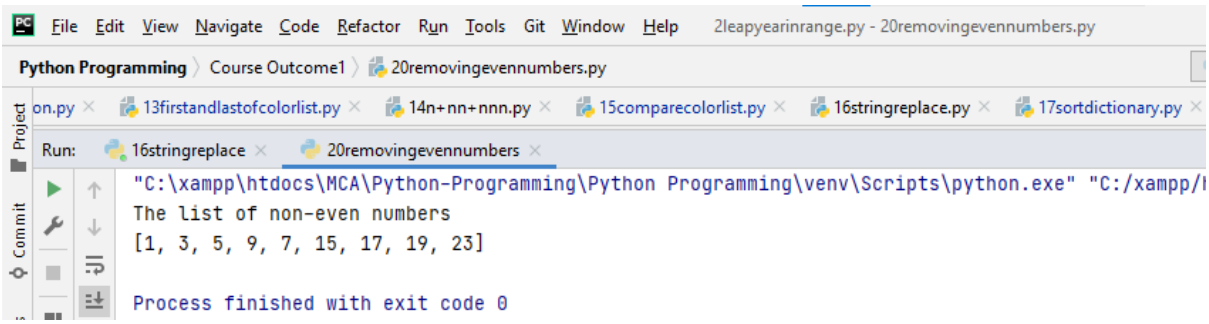
Step 3: Print list1

PROGRAM CODE :

20removingevennumbers.py	list1=[]
	list=[1,2,3,5,9,7,8,16,15,17,19,23,26]
	for i in range(len(list)):
	  if(list[i]%2==1):
	  list1.append(list[i])
	print("The list of non-positive numbers")
	print(list1)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 20 :FACTORIAL OF A NUMBER

AIM : Program to find the factorial of a number.

ALGORITHM :

Step 1: Read n.

Step 2: If n=0 go to step 8 else go to step 3.

Step 3: Go to function named factorial .set fact=1 and counter variable i=1.

Step 4: If i<n+1 go to step 5 otherwise go to step 7.

Step 5: Calculate fact=fact*i

Step 6 :Increment i and go to step 4.

Step 7 :Print fact as factorial.

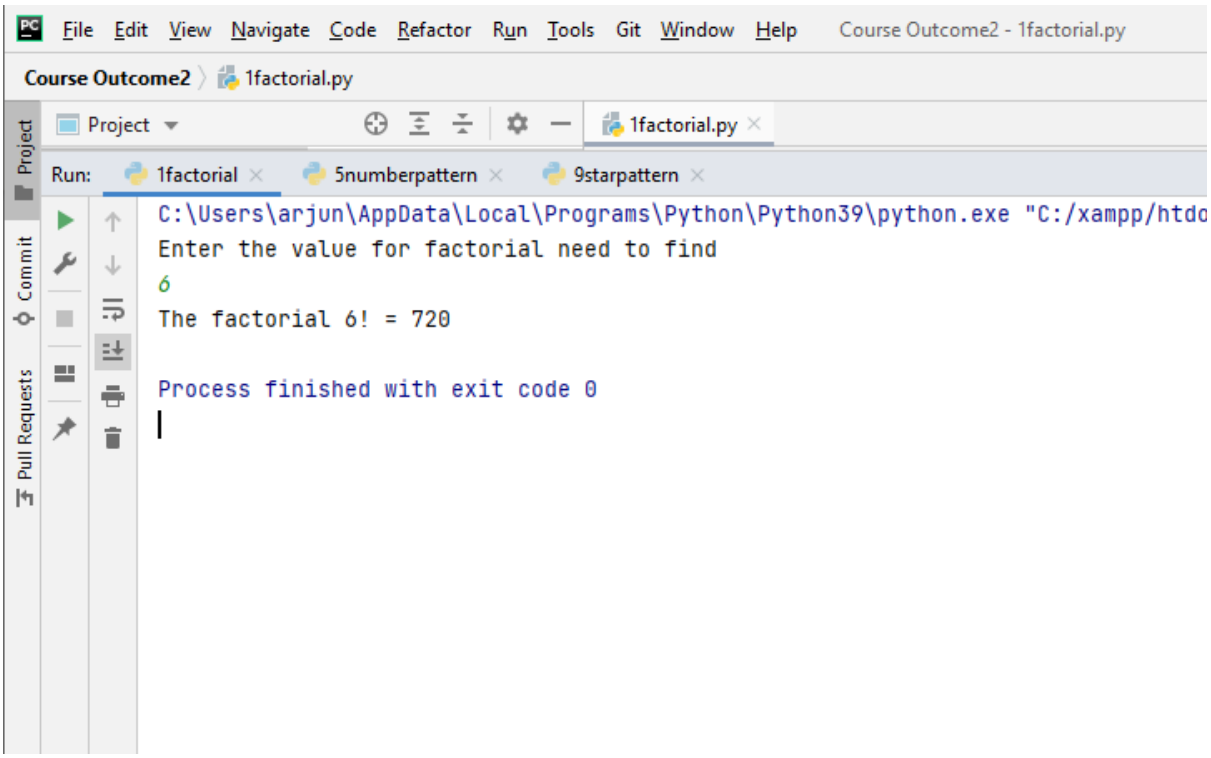
Step8 : Print factorial of 0 as 1.

PROGRAM CODE :

1factorial.py	def factorial(x):
	  fact=1
	  for i in range(1,x+1):
	  fact=i*fact
	  return fact
	n=int(input("Enter the value for factorial need to find\n"))
	if n==0:
	  print("The factorial 0! = 1")
	else:
	  print("The factorial "+str(n)+"! = "+str(factorial(n)))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 21 :FIBONACCI SERIES

AIM : Generate Fibonacci series of N terms.

ALGORITHM :

Step 1: Read n

Step 2: Set f1=0,f2=1 and counter variable i=0

Step 3:If i<=n go to step 4 else go to step 8

Step 4: if i=0, print f1.

Step 5 :if i=1 ,print f2.

Step 6 : If i>1 ,f3=f1+f2,print f3.

Step 7 : Increment counter variable i and go to step 3.

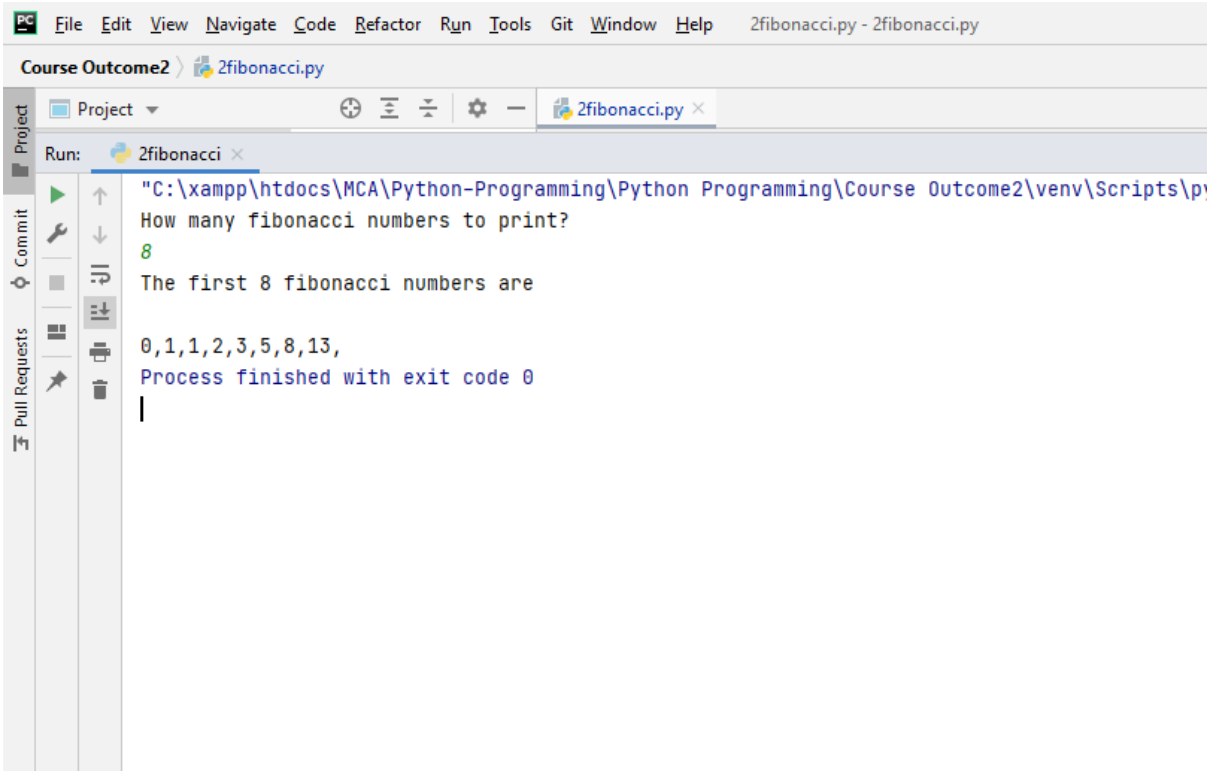
Step 8 : Stop.

PROGRAM CODE :

2fibonacci.py	n=int(input("How many fibonacci numbers to print?\n"))
	f1=0
	f2=1
	print("The first "+str(n)+" fibonacci numbers are\n")
	for i in range(n):
	  if i==0:
	  print(str(f1),end=",")
	  if i==1:
	  print(str(f2),end=",")
	  if i>1:
	  f3=f1+f2
	  print(str(f3),end=",")
	  f1=f2
	  f2=f3

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 22 :SUM OF LIST ITEMS

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

ALGORITHM :

Step 1 : Declaring a list with elements.

Step 2 : Set sum=0 and counter variable i =0

Step 3: If i less than size of the declared list ,go to step 4.

Step 4 : Calculate sum=sum+list[i],Print sum.

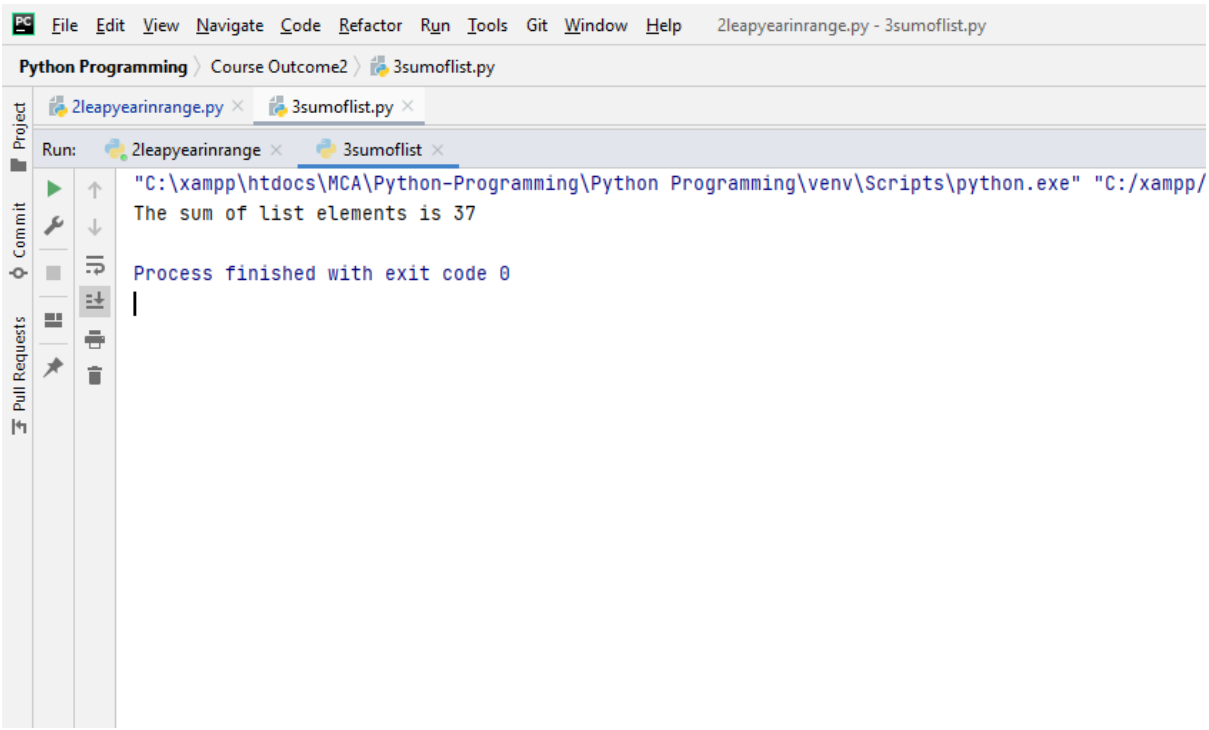
PROGRAM CODE :

3sumoflist.py	list=[5,6,7,8,9,2] sum=0 for i in range(len(list)): sum=sum+list[i] print("The sum of list elements is "+str(sum))

RESULT :

The above program is successfully executed and obtained the output.

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 3sumoflist.py
Python Programming
Course Outcome2
3sumoflist.py
2leapyearinrange.py
3sumoflist.py
Run:
2leapyearinrange
3sumoflist
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/
The sum of list elements is 37
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 23 :GENERATING NUMBERS

AIM : Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

ALGORITHM :

Step 1: Read lower limit and upper limit

Step 2 : Calculate roots of Upper limit and Lower limit.

Step 3 : Set counter variable i=root1.

Step 4 : If i<root2,go to step 5.Else go to step 11

Step 5 : Calculate, Square=i*i ,set x=square .Calculate rem=square%10

Step 6 : If rem=4 or rem=0 , Calculate square = square // 10, rem = square % 10 and go to step 6 ,else go to step 10

Step 6 : If rem=0 or rem=2 or rem=4 or rem=6 or rem=8 , Calculate square = square // 10, rem = square % 10 and go to step 7,else go to step 10 .

Step 7 : If rem=0 or rem=2 or rem=4 or rem=6 or rem=8 , Calculate square = square // 10, rem = square % 10 and go to step 8,else go to step 10 .

Step 8 : If rem=0 or rem=2 or rem=4 or rem=6 or rem=8 , Calculate square = square // 10, rem = square % 10 and go to step 9,else go to step 10.

Step 9 : If rem=0 or rem=2 or rem=4 or rem=6 or rem=8 , print x and go to step 10.

Step 10 : Increment Counter Variable and go to step 4

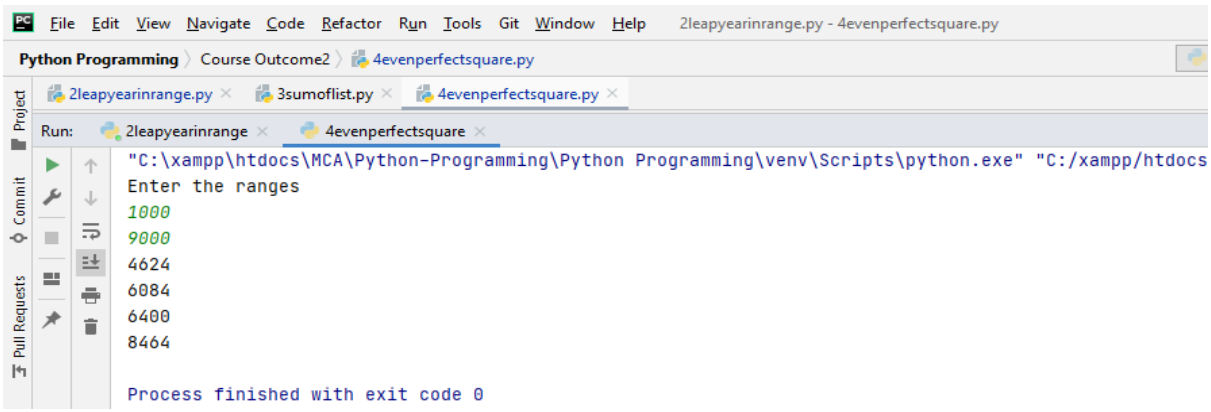
Step 11 : Stop.

PROGRAM CODE :

4evenperfectsquare.py	range1=int(input("Enter the ranges\n"))
	range2=int(input())
	root1=int(range1**(1/2))
	root2=int(range2**(1/2))
	for i in range(root1,root2):
	  x=square=i**2
	  rem=square%10
	  if rem==4 or rem==0:
	  square = square // 10
	  rem=square%10
	  if rem==0 or rem==2 or rem==4 or rem==6 or rem==8:
	  square = square // 10
	  rem = square % 10
	  if rem==0 or rem==2 or rem==4 or rem==6 or rem==8:
	  square = square // 10
	  rem = square % 10
	  if rem==0 or rem==2 or rem==4 or rem==6 or rem==8:
	  square = square // 10
	  rem = square % 10
	  if rem==0 or rem==2 or rem==4 or rem==6 or rem==8:
	  print(x)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 24 :NUMBER PYRAMID

AIM : Display the given pyramid with step number accepted from user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

ALGORITHM :

Step 1: Read n.

Step 2: Set counter variables i,j as 1.

Step 3: If i<n+1 go to step 4.else go to step 9

Step 4: If j<n+1,calculate x=i*j and go to step 5 ,else go to step 8.

Step 5: If j<=i print x, else print a space.Then Set x=x*j.

Step 6: Increment counter variable j and go to step 4.

Step 7: Increment counter variable i and go to step 3.

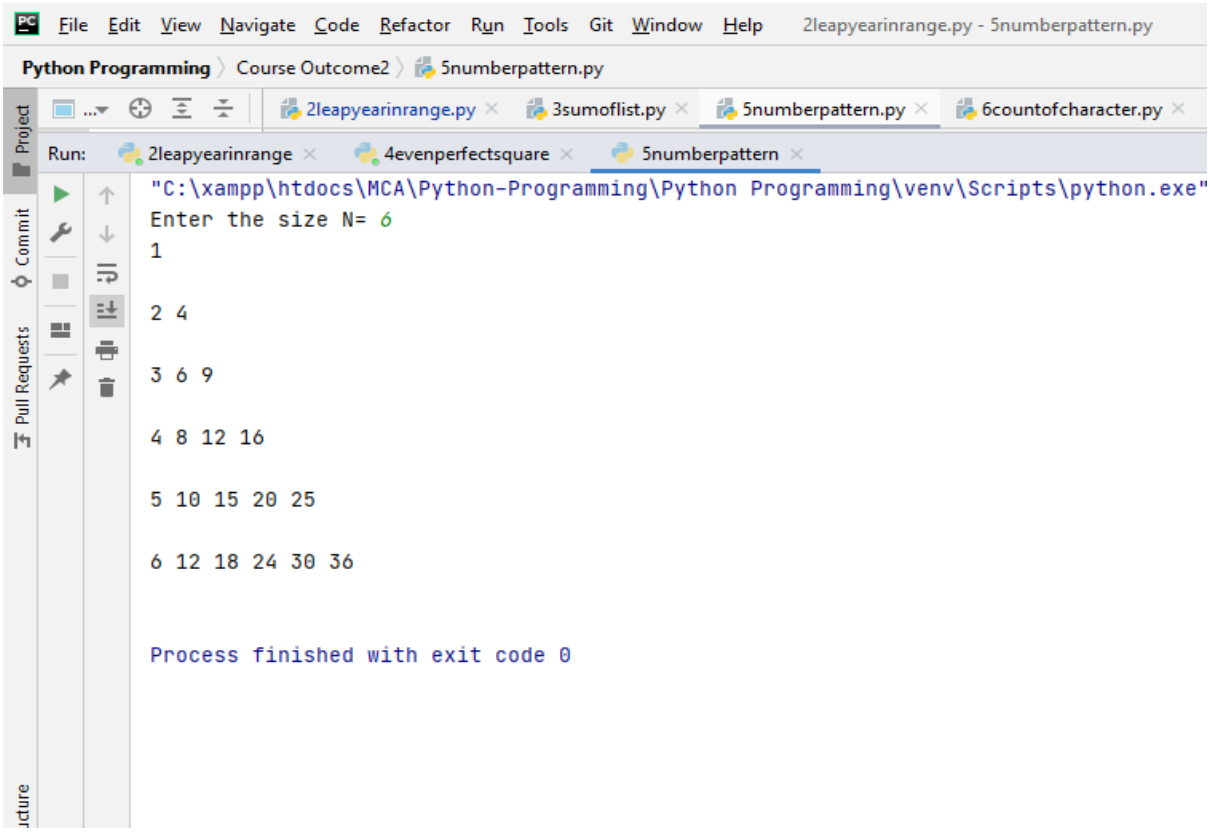
Step 8: Stop

PROGRAM CODE :

5numberpattern.py	n=int(input("Enter the size N= "))
	for i in range(1,n+1):
	  for j in range(1,n+1):
	  x=i*j
	  if j<=i:
	  print(str(x)+" ",end="")
	  else:
	  print(" ",end="")
	  x=x*j
	  print("\n")

RESULT : The above program is successfully executed and obtained the output.

OUTPUT :



PROGRAM 25 :CHARACTER FREQUENCY

AIM : Count the number of characters (character frequency) in a string.

ALGORITHM :

Step 1: Get string from the user.

Step 2: Initialise a empty dictionary.

Step 3: Use a for loop to iterate the letters of the word one by one.

Step 4: Compare if the letter already inside the dictionary ,increment its count.

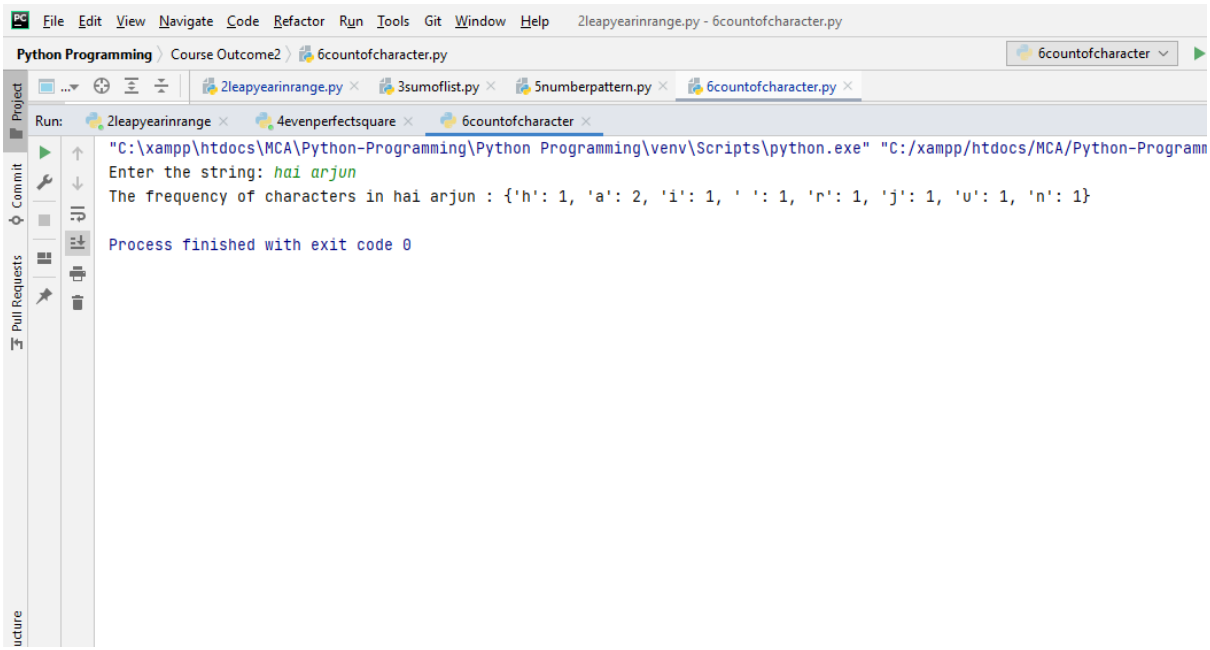
Step 5: Else add the letter to the dictionary with count one.

PROGRAM CODE :

6countofcharacter.py	word=str(input("Enter the string: "))
	freq={}
	
	for letter in word:
	  if letter in freq:
	  freq[letter]+=1
	  else:
	  freq[letter]=1
	print("The frequency of characters in",word,":",str(freq))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 6countofcharacter.py
Python Programming
Course Outcome2
6countofcharacter.py
6countofcharacter
2leapyearinrange.py
3sumoflist.py
5numberpattern.py
6countofcharacter.py
Run:
2leapyearinrange
4evenperfectsquare
6countofcharacter
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/htdocs/MCA/Python-Programm
Enter the string: hai arjun
The frequency of characters in hai arjun : {'h': 1, 'a': 2, 'i': 1, ' ': 1, 'r': 1, 'j': 1, 'u': 1, 'n': 1}
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 26 :STRING MODIFICATION

AIM : Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

ALGORITHM :

Step 1: Read a string.

Step 2: If last three letters of string is ”ing” go to step step 3,else go to step 4.

Step 3: Concatenate “ly” with string that without last “ing” part at end.

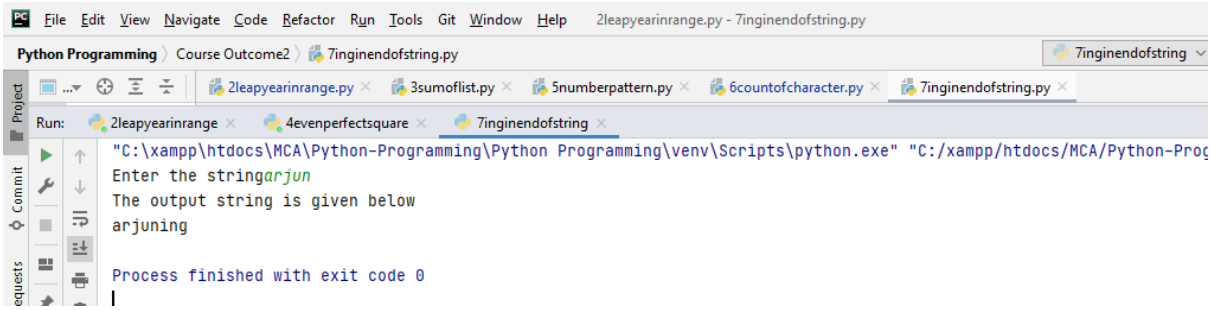
Step 4: Concatenate “ing” at end of the string.

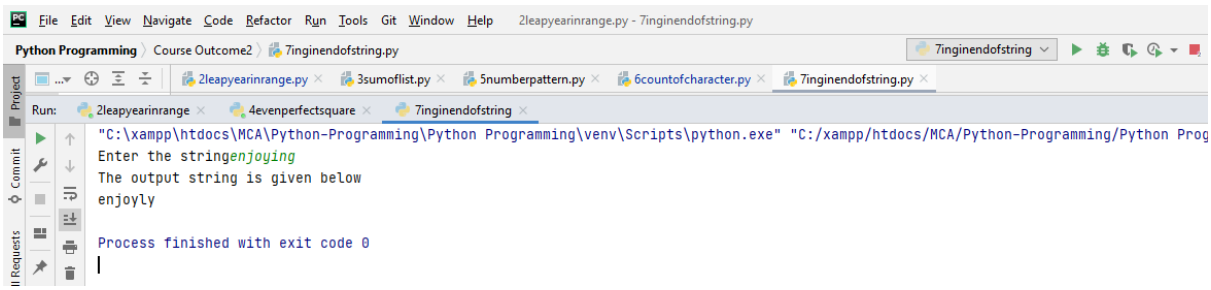
PROGRAM CODE :

7inginendofstring.py	string=input("Enter the string")
	print("The output string is given below")
	if string[-3:]=="ing":
	  print(string[:-3]+"ly")
	else:
	  print(string+"ing")

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help 2leapyearinrange.py - 7inginendofstring.py
Python Programming 〉 Course Outcome2 〉 7inginendofstring.py
7inginendofstring
2leapyearinrange.py
3sumoflist.py
5numberpattern.py
6countofcharacter.py
7inginendofstring.py
Run:
2leapyearinrange
4evenperfectsquare
7inginendofstring
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/htdocs/MCA/Python-Prog
Enter the stringarjun
The output string is given below
arjuning
Process finished with exit code 0


File Edit View Navigate Code Refactor Run Tools Git Window Help 2leapyearinrange.py - 7inginendofstring.py
Python Programming 〉 Course Outcome2 〉 7inginendofstring.py
7inginendofstring
2leapyearinrange.py
3sumoflist.py
5numberpattern.py
6countofcharacter.py
7inginendofstring.py
Run:
2leapyearinrange
4evenperfectsquare
7inginendofstring
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/htdocs/MCA/Python-Programming/Python Prog
Enter the stringenjoying
The output string is given below
enjoyly
Process finished with exit code 0

PROGRAM 27 :LENGTH OF LONGEST WORD

AIM : Accept a list of words and return length of longest word.

ALGORITHM :

Step 1: Read numbers strings want to enter.

Step 2: Read Strings

Step 3: Store size of each strings to list .

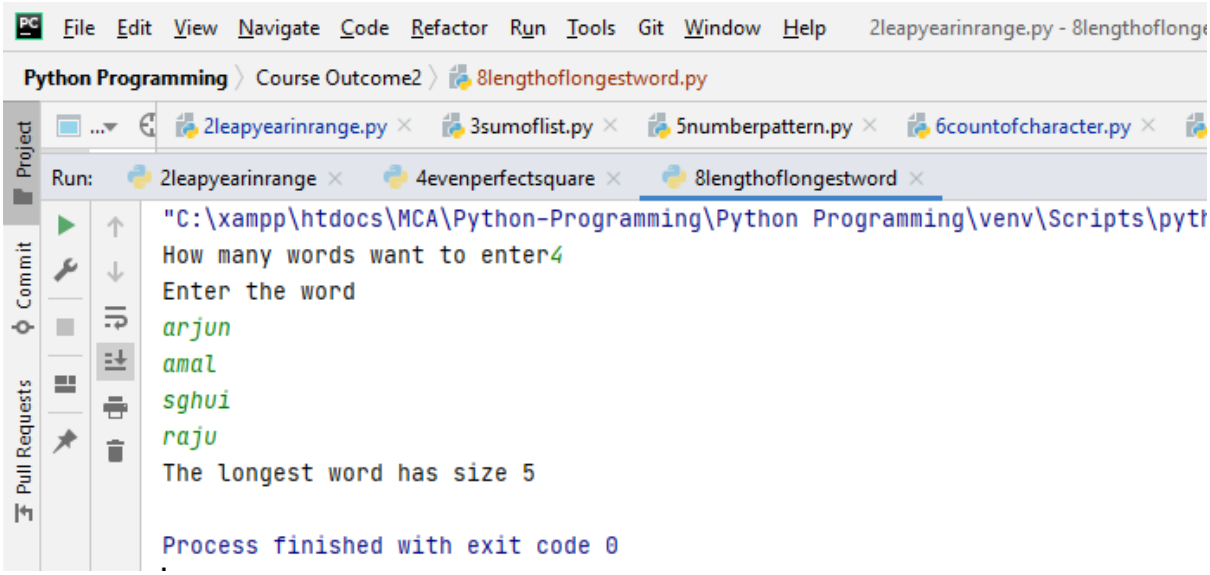
Step 4: Print maximum value in the list as the length of longest word.

PROGRAM CODE :

8lengthoflongestword.py	list=[] n=int(input("How many words want to enter")) print("Enter the word") for i in range(n): list.append(len(input())) print("The longest word has size "+str(max(list)))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 8lengthoflonge
Python Programming
Course Outcome2
8lengthoflongestword.py
2leapyearinrange.py
3sumoflist.py
5numberpattern.py
6countofcharacter.py
Run:
2leapyearinrange
4evenperfectsquare
8lengthoflongestword
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\pyth
How many words want to enter4
Enter the word
arjun
amal
sghui
raju
The longest word has size 5
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 28 : PATTERN WITH NESTED LOOP

AIM : Construct following pattern using nested loop.

```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *
*  *  *
*  *
*
```

ALGORITHM :

Step 1: Set counter variables i=j=0

Step 2: If i<9 go to step 3.else go to step 9.

Step 3: If j<5 go to step 4,else go to step 8.

Step 4: If i<5 go to step 5,else go to step 6.

Step 5: If j<=i, print "*".Else print " ".

Step 6: If if i+j<=8, print "*".Else print " ".

Step 7: Increment counter variable j and go to step 3.

Step 8: Increment counter variable i and Print new line,then go to step2.

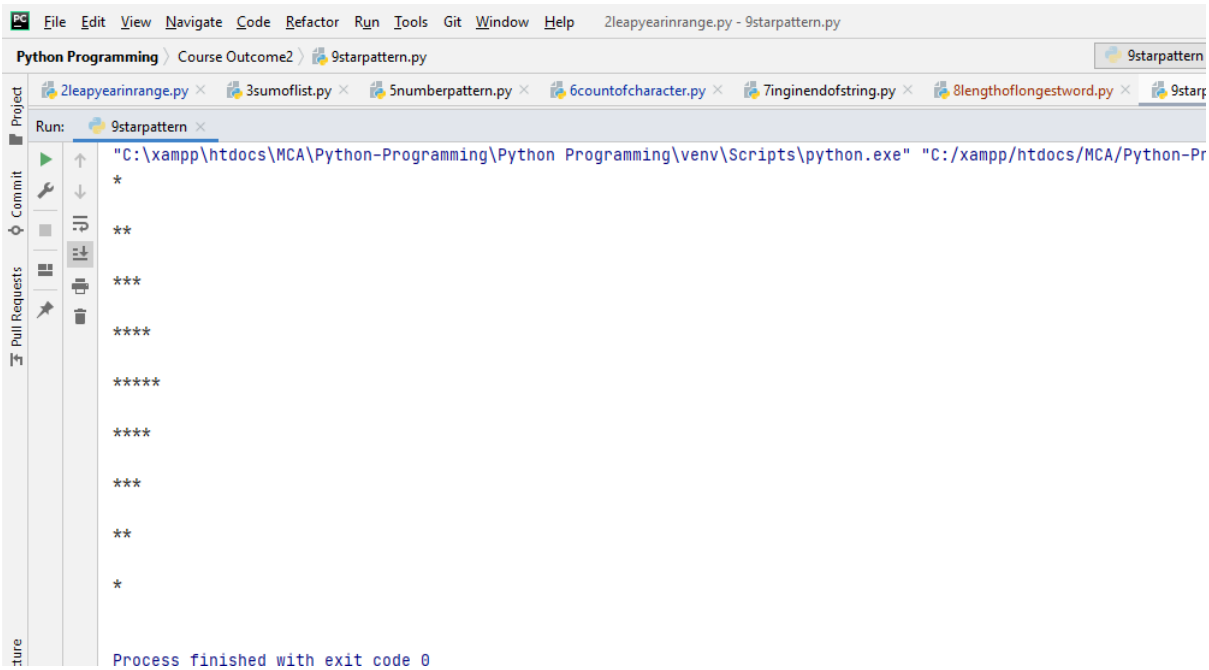
Step 9: Stop.

PROGRAM CODE :

9starpattern.py	for i in range(0,9):
	  for j in range(0,5):
	  if i < 5:
	  if j<=i:
	  print("*",end="")
	  else:
	  print(" ",end="")
	  else:
	  if i+j<=8:
	  print("*",end="")
	  else:
	  print(" ",end="")
	  print("\n")

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 29 :FACTORS OF A NUMBER

AIM : Generate all factors of a number.

ALGORITHM :

Step 1: Declare Empty list and read number.

Step 2: Initialise counter variable i=1.

Step 3: If i<((number/2)+1) go to step 4 ,else go to step 6.

Step 4: If number%i=0 ,add value of i to the list.

Step 5: Increment counter variable i and go to step 3.

Step 6: Add number to the list.

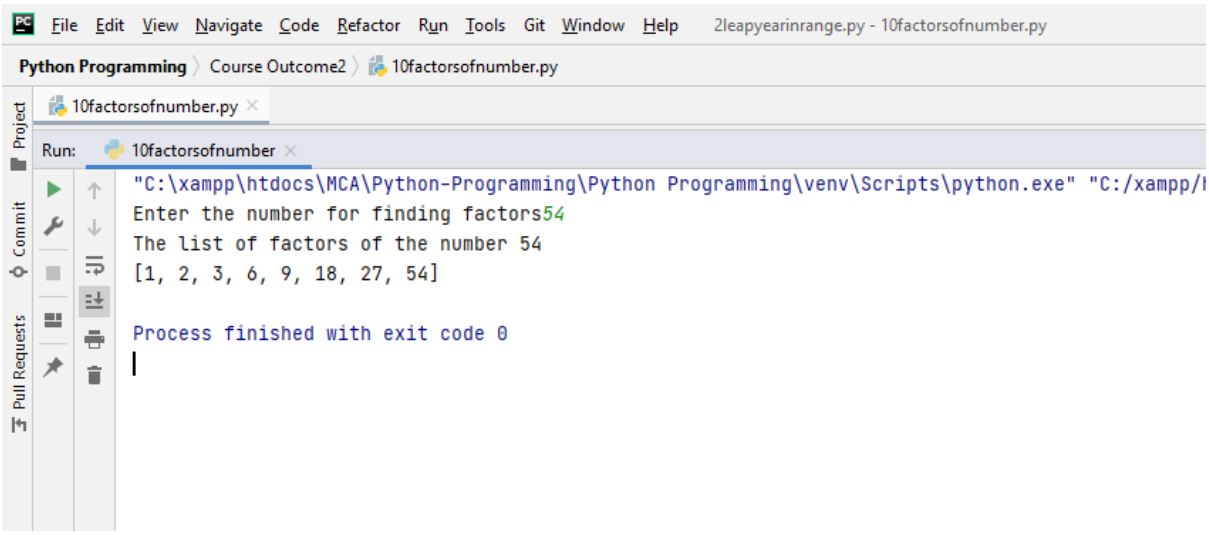
Step 7: Print the list as factors of number.

PROGRAM CODE :

10factorsofnumber.py	list=[]
	number=int(input("Enter the number for finding factors"))
	for i in range(1,int((number/2)+1)):
	  if number%i==0:
	  list.append(i)
	list.append(number)
	print("The list of factors of the number "+str(number))
	print(list)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - 10factorsofnumber.py
Python Programming
Course Outcome2
10factorsofnumber.py
Project
Commit
Pull Requests
Run:
10factorsofnumber
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/
Enter the number for finding factors54
The list of factors of the number 54
[1, 2, 3, 6, 9, 18, 27, 54]
Process finished with exit code 0

PROGRAM 30 :LAMBDA FUNCTIONS

AIM : Write lambda functions to find area of square, rectangle and triangle

ALGORITHM :

Step 1: Declare and define an anonymous function using lambda keyword.

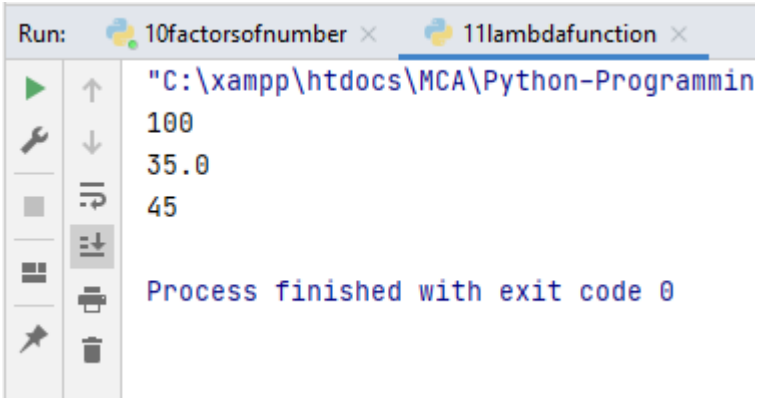
Step 2: Implement above declared lambda function.

PROGRAM CODE :

11lambdafunction.py	square = lambda s : s*s rectangle = lambda l,b : l*b triangle = lambda b,h : (b*h)/2 print(square(10)) print(triangle(7,10)) print(rectangle(9,5))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 31 : WORK WITH BUILT-IN PACKAGES

AIM : Work with built-in packages

ALGORITHM :

Step 1: Import math library for math related functions.

Step 2: Call and print an instance of factorial function from math library and print it.

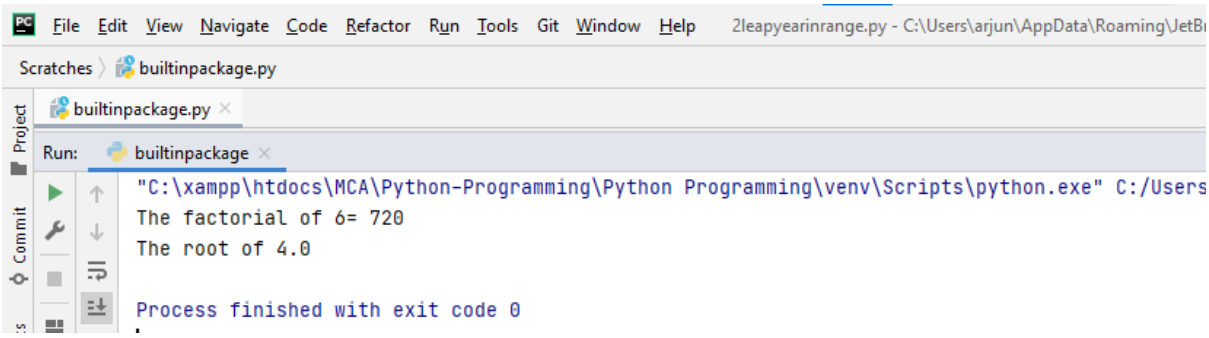
Step 3: Call and print an instance of square root function from math library and print it.

PROGRAM CODE :

builtinpackage.py	import math
	#Return factorial of a number
	print("The factorial of 6=",math.factorial(6))
	
	# Print the square root of different numbers
	print ("The root of",math.sqrt(16))

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 32 : WORKING WITH MODULE

AIM : Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements.

ALGORITHM :

Step1: Create a folder with name ‘graphics’ , add ‘_init_.py’ file in it.

Step2: Create a subfolder ‘graphics3D’ in ‘graphics’ add ‘_init_.py’ file in it.

Step3: Create python modules for circle & rectangle, write functions for calculating area & perimeter for each in ‘graphics’ folder.

Step4: Create python modules for sphere & cuboid, write functions for calculating area & perimeter for each in ‘graphics3D’ folder.

Step5 : Access the functions in the above modules and call the function using different types of imports.

PROGRAM CODE :

main.py	import graphics.rectangle
	from graphics.circle import *
	from graphics.graphics3D import cuboid
	import graphics.graphics3D.sphere
	print("What do you want to find?")
	print(
	"1.Area of Rectangle\n2.Perimeter of Rectangle\n3.Area and Perimeter of rectangle\n4.Area of Circle\n5.Perimeter of Circle\n6.Area and Perimeter of Circle\n7.Area of Cuboid\n8.Perimeter of Cuboid\n9.Area and Perimeter of Cuboid\n10.Area of Sphere\n11.Perimeter of Sphere\n12.Area and Perimeter of Sphere")
	option = int(input("Enter the Option"))
	if option == 1:
	a = int(input("Enter Length and Width"))
	b = int(input())
	print("Area of Rectangle=" + str(Graphics.rectangle.arear(a, b)))
	elif option == 2:
	a = int(input("Enter Length and Width"))
	b = int(input())

```
print("The Perimeter of the Rectangle=" + str(Graphics.rectangle.perimeterr(a,
b)))
elif option == 3:
   a = int(input("Enter Length and Width"))
   b = int(input())
   print("Area of Rectangle=" + str(Graphics.rectangle.arear(a, b)))
   print("The Perimeter of the Rectangle=" +
str(Graphics.rectangle.perimeterr(a, b)))
elif option == 4:
   r = int(input("Enter the radius of Circle "))
   print("The Area of Circle=" + str(areacir(r)))
elif option == 5:
   r = int(input("Enter the radius of Circle "))
   print("The Perimeter of the Circle=" + str(perimetercir(r)))
elif option == 6:
   r = int(input("Enter the radius of Circle "))
   print("The Area of Circle=" + str(areacir(r)))
   print("The Perimeter of the Circle=" + str(perimetercir(r)))
elif option == 7:
   a = int(input("Enter the length width hight of cuboid"))
   b=int(input())
   h=int(input())
   print("The Area of Cuboid=" + str(cuboid.areacu(a, b, h)))
elif option==8:
   a = int(input("Enter the length width hight of cuboid"))
   b = int(input())
   h = int(input())
   print("The Perimeter of the Cuboid="+str(cuboid.perimetecu(a,b,h)))
elif option==9:
   a = int(input("Enter the length width hight of cuboid"))
   b=int(input())
   h=int(input())
   print("The Area of Cuboid=" + str(cuboid.areacu(a, b, h)))
   print("The Perimeter of the Cuboid="+str(cuboid.perimetecu(a,b,h)))
elif option==10:
   r=int(input("Enter the radius of Sphere"))
   print("The Area of Sphere="+str(Graphics.Graphics3D.sphere.areasp(r)))
elif option==11:
   r=int(input("Enter the radius of Sphere"))
   print("The Perimeter of
Sphere="+str(Graphics.Graphics3D.sphere.perimetesp(r)))
elif option==12:
   r=int(input("Enter the radius of Sphere"))
   print("The Area of Sphere="+str(Graphics.Graphics3D.sphere.areasp(r)))
   print("The Perimeter of
Sphere="+str(Graphics.Graphics3D.sphere.perimetesp(r)))
```

	else: print("Choose only Above Options")

circle.py	def areacir(r):
	result=3.14*(r*r)
	return result
	def perimetercir(r):
	result=4*3.14*r
	return result

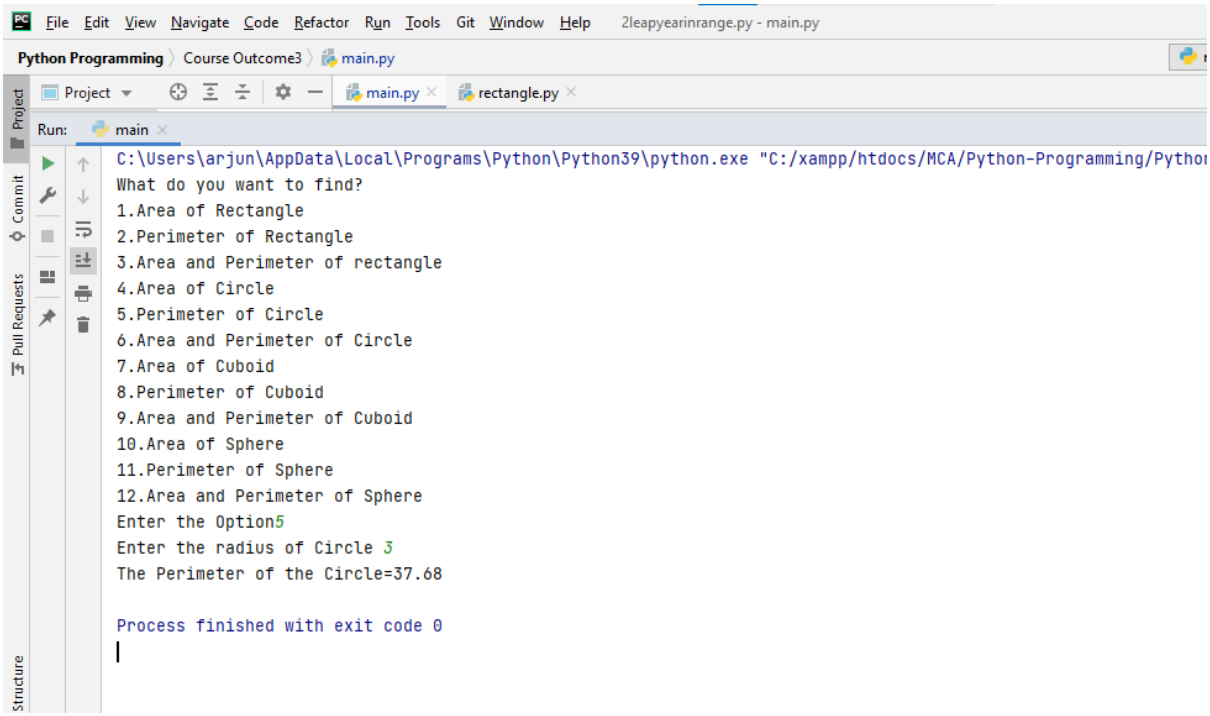
rectangle.py	def arear(a,b):
	result=a*b
	return result
	def perimeterr(a,b):
	result=(a+b)*2
	return result

cuboid.py	def areacu(a,b,h):
	result=2*((a*b)+(b*h)+(h*a))
	return result
	def perimetecu(a,b,h):
	result=(a+b+h)*4
	return result

sphere.py	def areasp(r):
	result=4*3.14*r*r
	return result
	def perimetesp(r):
	result=(2*3.14*r)
	return result

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 33 : FIND AREA OF RECTANGLES AND COMPARE THEM.

AIM : Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

ALGORITHM :

Step 1: Define a class rectangle with a constructor and functions area and perimeter.

Step 2: Accept length and breath of two rectangles from the user.

Step 3: Create two objects

Step 4: Compare the area of two objects.

PROGRAM CODE :

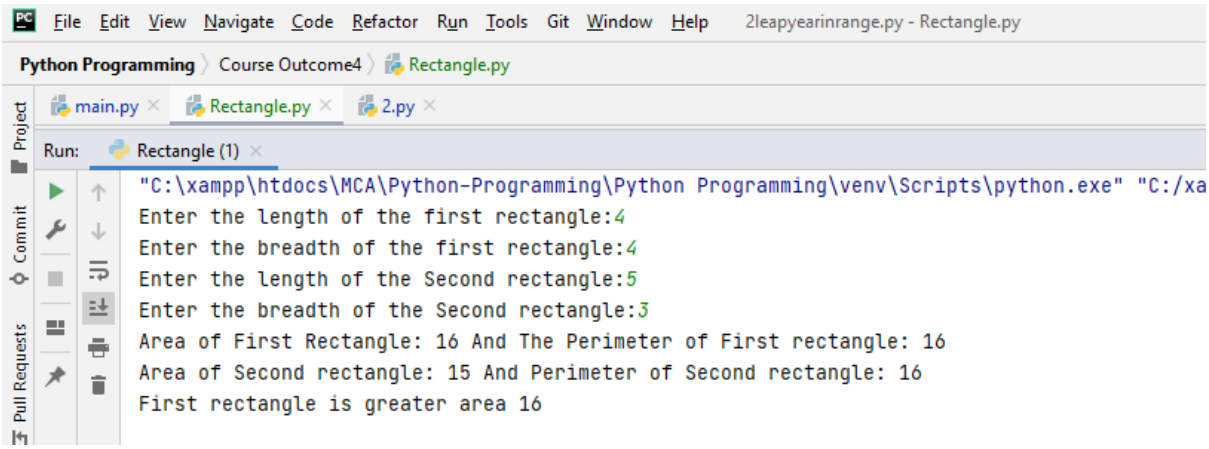
Rectangle.py

```
class rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        a = self.length * self.breadth
        return a

    def perimeter(self):
        b = 2 * (self.length + self.breadth)
        return b
a = int(input("Enter the length of the first rectangle:"))
b = int(input("Enter the breadth of the first rectangle:"))
c = int(input("Enter the length of the Second rectangle:"))
d = int(input("Enter the breadth of the Second rectangle:"))
obj1 = rectangle(a, b)
obj2 = rectangle(c, d)
print("Area of First Rectangle:", obj1.area(), "And", "The Perimeter of First
rectangle:", obj1.perimeter())
print("Area of Second rectangle:", obj2.area(), "And", "Perimeter of Second
rectangle:", obj2.perimeter())

if obj1.area() == obj2.area():
```

```
    print("Both  rectangle have same area ", obj1.area())
elif obj1.area() > obj2.area():
    print("First rectangle is greater area", obj1.area())
else:
    print("Second rectangle is greater area ", obj2.area())
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 34 : BANK TRANSACTIONS

AIM : Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

ALGORITHM :

Step 1: Create a class bank account

Step 2: Define a constrctor and three functions display,deposit,withdraw.

Step 3: Create objects.

Step 4: Call the functions and perform the corresponding operations.

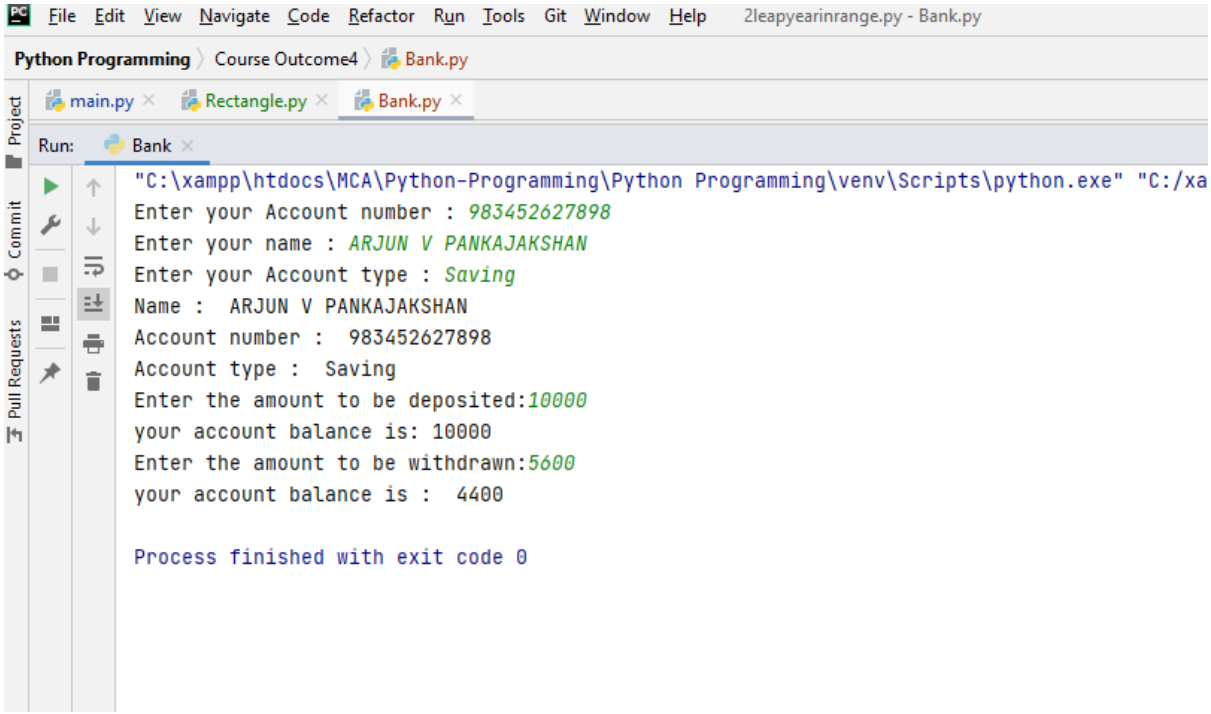
PROGRAM CODE

Bank.py

```
class bankAccount:
    def __init__(self):
        self.balance=0
        self.accountnumber=int(input("Enter your Account number : "))
        self.name=input("Enter your name : ")
        self.accounttype=input("Enter your Account type : ")
    def display(self):
        print("Name : ",self.name)
        print("Account number : ",self.accountnumber)
        print("Account type : ",self.accounttype)
    def deposit(self):
        amount=int(input("Enter the amount to be deposited:"))
        self.balance+=amount
        print("your account balance is:",self.balance)
    def withdraw(self):
        amount=int(input("Enter the amount to be withdrawn:"))
        if(amount>self.balance):
            print("INSUFFICIENT BALANCE")
        else:
            self.balance-=amount
            print("your account balance is : ",self.balance)
account=bankAccount()
account.display()
account.deposit()
account.withdraw()
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 35 : '<' OPERATOR OVERLOADING

AIM : Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

ALGORITHM :

Step 1: Define a class rectangle with private attributes length,breadth,area.

Step 2: Define constructor and function for calculating area

Step 3: Use the __gt__ function to overload the greater than operator and write the logic for comparing the areas.

Step 4: Create two objects and call the functions area and rectangle.

Step 5: Compare the two objects and print the corresponding output.

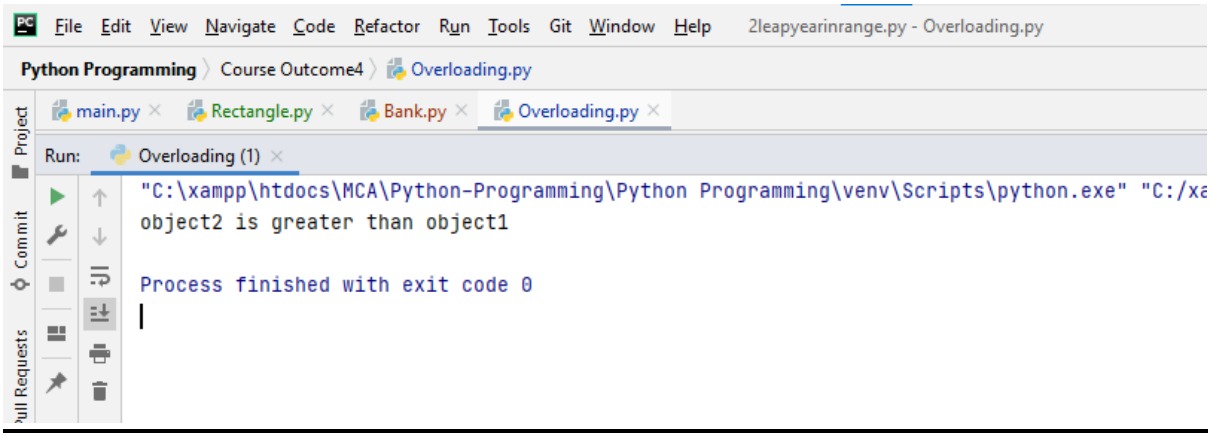
PROGRAM CODE :

Overloading.py

```
class rectangle:
    __length=0
    __breadth=0
    __area=0
    def __init__(self,l,w):
        self.__length=l
        self.__width=w
    def area(self):
        self.__area=self.__length*self.__width
    def __gt__(self,other):
        if(self.__area>other.__area):
            return True
        else:
            return False
ob1=rectangle(5,4)
ob1.area()
ob2=rectangle(6,7)
ob2.area()
if(ob1>ob2):
    print("object1 is greater than object2")
else:
    print("object2 is greater than object1")
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 36 : '+' OPERATOR OVERLOADING

AIM : Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

ALGORITHM :

Step 1: Define a class time with a constructor, and function naming time.

Step 2: Use __add__function to overload '+'.

Step 3: Create two objects and call the function Time.

Step 4: Print the corresponding output.

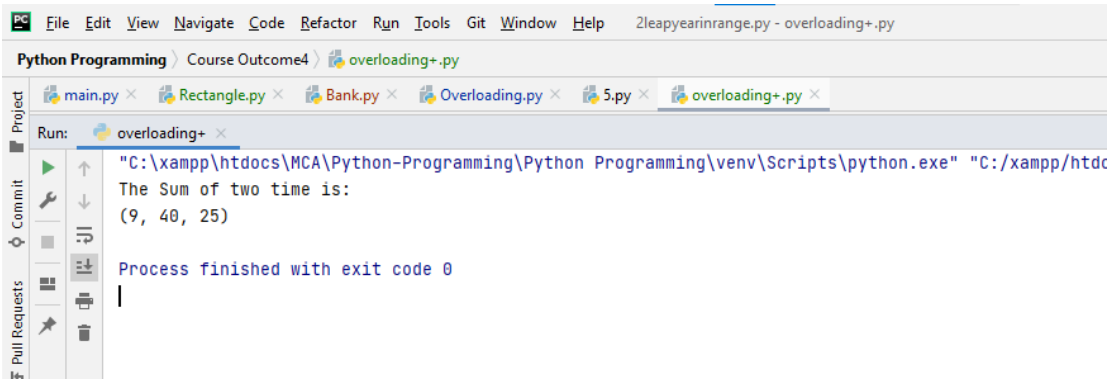
PROGRAM CODE :

overloading+.py

```
class Time:
    def __init__(self,h,m,s):
        self.__hour=h;
        self.__minute=m
        self.__seconds=s

    def time(self):
        if self.__seconds>=60:
            self.__seconds-=60
            self.__minute+=60
        if self.__minute>=60:
            self.__minute-=60
            self.__hour+=1

    def __add__(self,other):
        self.__hour=self.__hour+other.__hour
        self.__minutr=self.__minute+other.__minute
        self.__seconds=self.__seconds+other.__seconds
        return(self.__hour,self.__minute,self.__seconds)
obj1=Time(2,40,80)
obj1.time()
obj2=Time(6,40,5)
obj2.time()
print("The Sum of two time is:")
print(obj1+obj2)
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 37 : METHOD OVERRIDING

AIM : Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no of pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

ALGORITHM :

Step 1: Define a class publisher .

Step 2: Derive classes book and python from the class publisher.

Step 3: Create an object and pass the values.

Step 4: Call the function display to render the output.

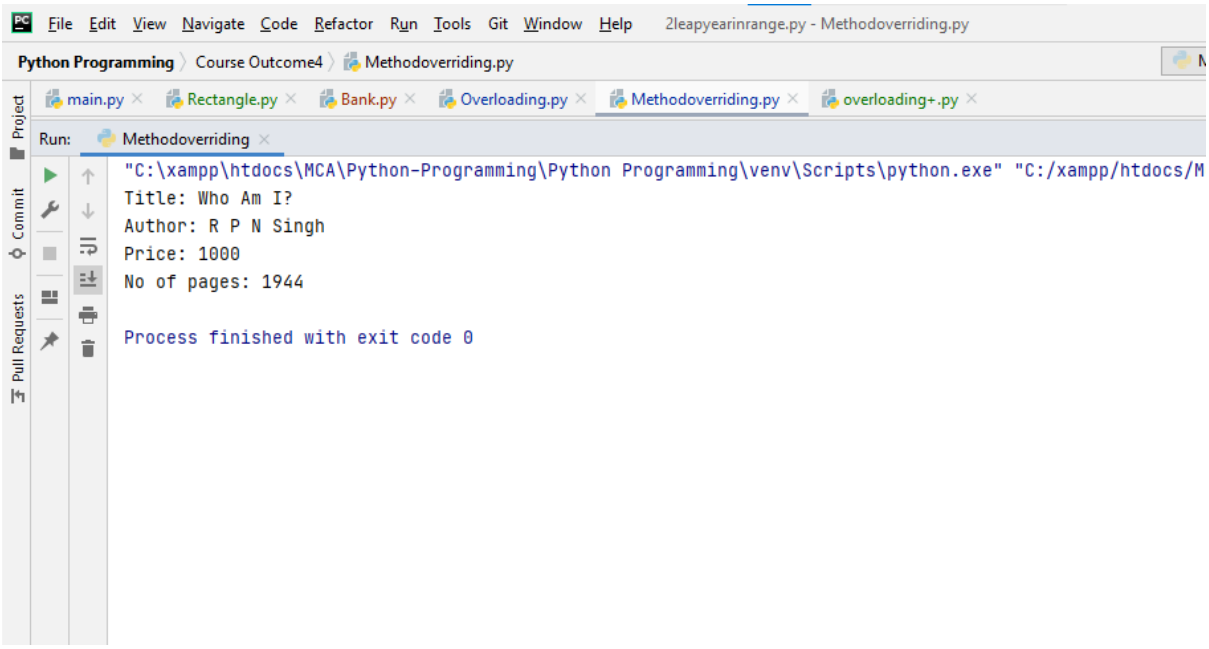
PROGRAM CODE :

Methodoverriding.py	class Publisher:
	  def __init__(self, Pubname):
	  self.Pubname = Pubname
	
	  def display(self):
	  print("Publisher name is:", self.Pubname)
	
	class Book(Publisher):
	  def __init__(self, Pubname, title, author):
	  Publisher.__init__(self, Pubname)
	  self.title = title
	  self.author = author
	
	  def display(self):
	  print("Title:", self.title)
	  print("Author:", self.author)
	
	class Python(Book):
	  def __init__(self, Pubname, title, author, price, no_of_pages):
	  Book.__init__(self, Pubname, title, author)

```
        self.price = price
        self.no_of_pages = no_of_pages

    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
        print("Price:", self.price)
        print("No of pages:", self.no_of_pages)


ob1 = Python("New coders", "Who Am I?", "R P N Singh ", 1000, 1944)
ob1.display()
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 38 : READING FILES LINE BY LINE

AIM : Write a Python program to read a file line by line and store it into a list

ALGORITHM :

Step1: Open a text file.

Step2: Write data to the file using File.write("data") and close the file

Step3: Then,open the file for read operation

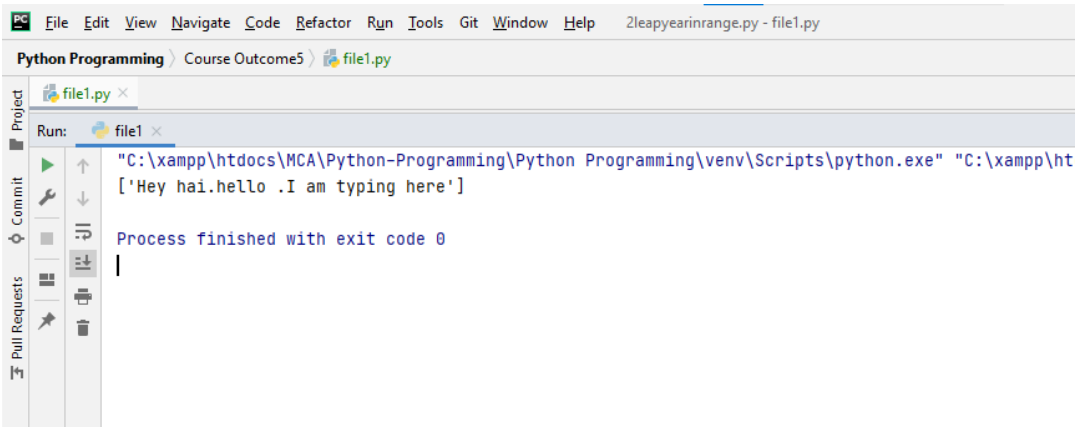
Step4: Print the contents of file as a list using File.readlines() and close the file

PROGRAM CODE :

file1.py	newFile = open("content.txt","a") newFile.write("Hey hai.hello .I am typing here ") newFile.close() readFile = open("content.txt","r") print(readFile.readlines()) readFile.close()

RESULT : The above program is successfully executed and obtained the output

OUTPUT :



PROGRAM 39 : COPY ODD LINES OF A FILE

AIM : Python program to copy odd lines of one file to other

ALGORITHM :

Step1: Open a file for write operation

Step2: Write data to the file using File.write("data") and close the file

Step3: Then, open the file for read operation and copy the contents of the file to a variable as a list and close the file

Step4: Create a new file object.

Step5: Iterate and copy old lines of the variable to the new file and close the new file.

PROGRAM CODE :

File2.py	newFile = open("content.txt","w") newFile.write("Arjun V P \nSemester 1 \nMCA Department \nTKMCE \nKollam") newFile.close() readFile = open("content.txt","r") lines = readFile.readlines() readFile.close() oddFile = open("oddcontent.txt","w") for i in range(0,len(lines),2): oddFile.write(lines[i]) oddFile.close()

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


content - Notepad
File Edit Format View Help
Arjun V P
Semester 1
MCA Department
TKMCE
Kollam


oddcontent - Notepad
File Edit Format View Help
Arjun V P
MCA Department
Kollam

PROGRAM 40 : READ CSV FILE

AIM : Write a Python program to read each row from a given csv file and print a list of strings.

ALGORITHM :

Step1: Import CSV module.

Step2: Open CSV file and write data to the rows of the CSV file.

Step3: Then, open the CSV file for read operation.

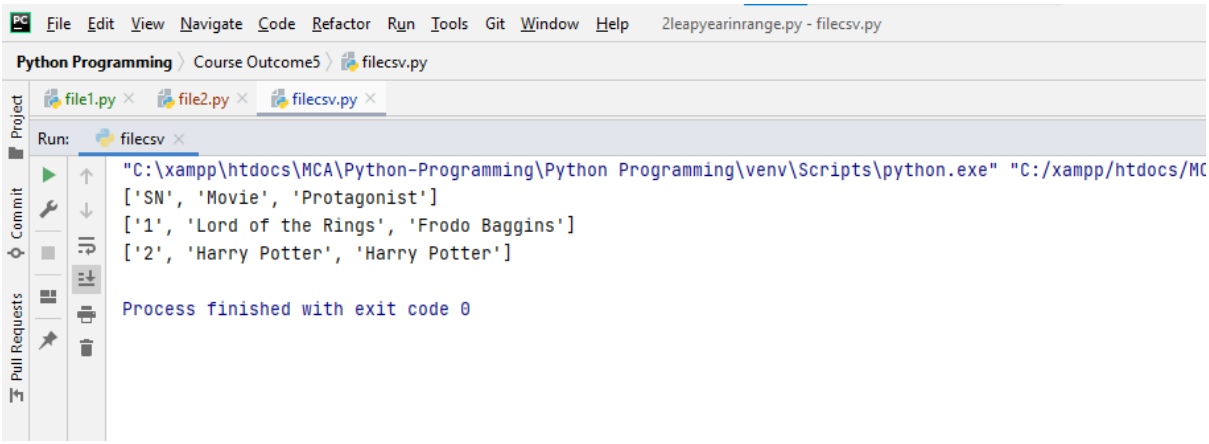
Step4: Iterate and print the rows of the CSV file.

PROGRAM CODE :

filecsv.py	import csv
	with open('arjun.csv', 'w', newline='') as file:
	  writer = csv.writer(file)
	  writer.writerow(["SN", "Movie", "Protagonist"])
	  writer.writerow([1, "Lord of the Rings", "Frodo Baggins"])
	  writer.writerow([2, "Harry Potter", "Harry Potter"])
	
	with open('arjun.csv', 'r') as file:
	  reader = csv.reader(file)
	  for row in reader:
	  print(row)

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - filecsv.py
Python Programming
Course Outcome5
filecsv.py
file1.py
file2.py
filecsv.py
Run:
filecsv
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/htdocs/MC
['SN', 'Movie', 'Protagonist']
['1', 'Lord of the Rings', 'Frodo Baggins']
['2', 'Harry Potter', 'Harry Potter']
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 41 : READ SPECIFIC COLUMNS OF A CSV FILE

AIM : Write a Python program to read specific columns of a given CSV file and print the content of the columns.

ALGORITHM :

Step1: Import CSV module.

Step2: Open CSV file and write data to the rows of the CSV file.

Step3: Then, open the CSV file for read operation.

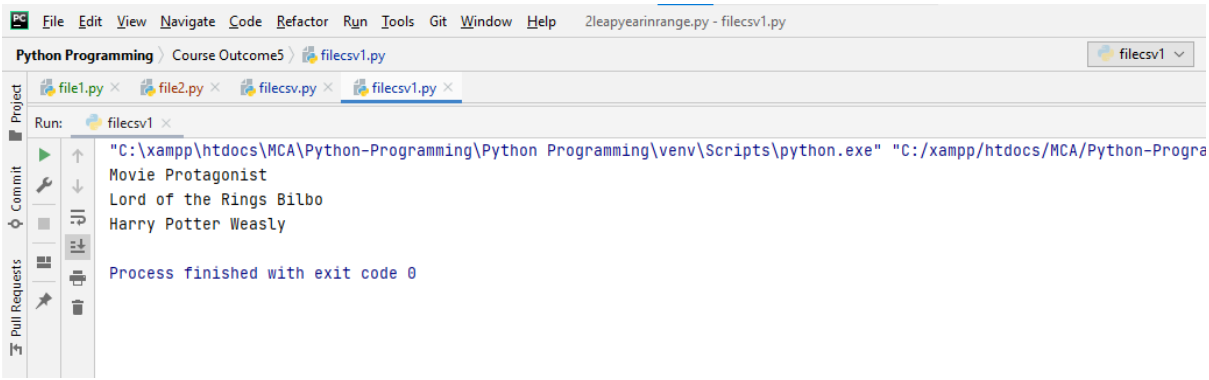
Step4: Iterate and print specified columns of the file and close the file.

PROGRAM CODE :

filecsv1.py	import csv
	with open('protagonist.csv', 'w', newline='') as file:
	  writer = csv.writer(file)
	  writer.writerow(["SN", "Movie", "Protagonist"])
	  writer.writerow([1, "Lord of the Rings", "Bilbo"])
	  writer.writerow([2, "Harry Potter", "Weasly"])
	
	with open('protagonist.csv', 'r') as file:
	  reader = csv.reader(file)
	  for row in reader:
	  print(row[1]+" "+row[2])

RESULT : The above program is successfully executed and obtained the output

OUTPUT :


File Edit View Navigate Code Refactor Run Tools Git Window Help
2leapyearinrange.py - filecsv1.py
Python Programming
Course Outcome5
filecsv1.py
filecsv1
file1.py
file2.py
filecsv.py
filecsv1.py
Run:
filecsv1
"C:\xampp\htdocs\MCA\Python-Programming\Python Programming\venv\Scripts\python.exe" "C:/xampp/htdocs/MCA/Python-Progra
Movie Protagonist
Lord of the Rings Bilbo
Harry Potter Weasly
Process finished with exit code 0
Project
Commit
Pull Requests

PROGRAM 42 : DICTIONARY TO A CSV FILE

AIM : Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

ALGORITHM :

Step1: Import CSV module.

Step3: Define columns of CSV file.

Step4: Declare a dictionary.

Step5: Use csv.DictWriter() to convert dictionary to a csv file directly

Step6: Open CSV file for read operations.

Step7: Iterate and print the rows of the CSV file.

PROGRAM CODE :

dictcsvfile.py

```
import csv
csv_columns = ['No','Name','Country']
dict_data = [
{'No': 1, 'Name': 'Sachin', 'Country': 'India'},
{'No': 2, 'Name': 'Broad', 'Country': 'England'},
{'No': 3, 'Name': 'Shri Ram', 'Country': 'India'},
{'No': 4, 'Name': 'Smith', 'Country': 'Australia'},
{'No': 5, 'Name': 'Yuva Raj', 'Country': 'India'},
]
csv_file = "names.csv"

with open(csv_file, 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames=csv_columns)
    writer.writeheader()
    for data in dict_data:
        writer.writerow(data)


with open('names.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

RESULT : The above program is successfully executed and obtained the output

OUTPUT :