**PROGRAM NO.: 1a**

**AIM:** To write and run Python instructions in Interactive interpreter and a Python Script

**THEORY:**

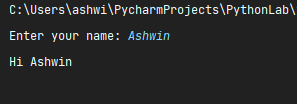
Interactive mode, also known as the REPL provides us with a quick way of running blocks or a single line of Python code. The code executes via the Python shell, which comes with Python installation. Interactive mode is handy when you just want to execute basic Python commands or you are new to Python programming and just want to get your hands dirty with this beautiful language.

**PROGRAM:**

name = input('Enter your name: ')

print(f'Hi {name}')

**OUTPUT:**



**PROGRAM NO.: 1b**

**AIM:** Write a Python program which accepts the radius of a circle from the user and compute the area.

**THEORY:**

Take input radius from user. Formula to calculate area is (3.14\*r\*r).

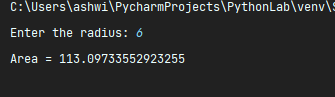
**PROGRAM:**

from math import pi

radius = float(input('Enter the radius: '))

print(f'Area = {pi \* radius \* radius}')

**OUTPUT:**



**PROGRAM NO.: 1c**

**AIM:** Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them.

**THEORY:**

Take two names as input from user in 2 different variables.

Print these 2 variables in reverse order.

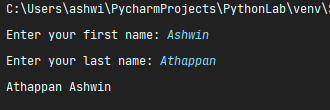
**PROGRAM:**

f\_name = input('Enter your first name: ')

l\_name = input('Enter your last name: ')

print(f'{l\_name} {f\_name}')

**OUTPUT:**



**PROGRAM NO.: 2a**

**AIM:** Write a Program for checking whether the given number is an even number or not.

**THEORY:**

Take a number as user input. If it (num%2==0) then it is even. Else it is odd.

**PROGRAM:**

num = int(input('Enter a number: '))

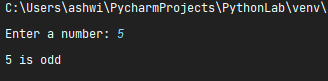
if num % 2 == 0:

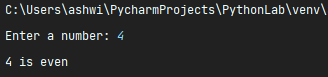
print(f'{num} is even')

else:

print(f'{num} is odd')

**OUTPUT:**





**PROGRAM NO.: 2b**

**AIM:** Write a Python program to find the median among three given numbers.

**THEORY:**

In Python, the statistics. median() function is used to calculate the median value of a data set. If the number of data points is odd, the middle data point will be returned. If the number is even, the median is the midpoint between the two middle values.

**PROGRAM:**

a = float(input("Input first number: "))

b = float(input("Input second number: "))

c = float(input("Input third number: "))

if a > b:

if a < c:

median = a

if b > c:

median = b

else:

median = c

else:

if a > c:

median = a

if b < c:

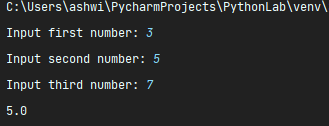
median = b

else:

median = c

print(median)

**OUTPUT:**



**PROGRAM NO.: 3a**

**AIM:** Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included).

**THEORY:**

If a number between 1500 and 2700 is exactly divisible by both 5 and 7 it should be printed.

**PROGRAM:**

for i in range(1500, 2701):

if i % 7 == 0 and i % 5 == 0:

print(i, end=" ")

**OUTPUT:**

C:\Users\ashwi\PycharmProjects\PythonLab\venv\Scripts\python.exe C:/Users/ashwi/PycharmProjects/PythonLab/Lab/3a.py

1505 1540 1575 1610 1645 1680 1715 1750 1785 1820 1855 1890 1925 1960 1995 2030 2065 2100 2135 2170 2205 2240 2275 2310 2345 2380 2415 2450 2485 2520 2555 2590 2625 2660 2695

Process finished with exit code 0

**PROGRAM NO.: 3b**

**AIM:** Write a Python program to guess a number between 1 to 9.

**THEORY:**

Import a library called Random. In random use a method called randint and specify the range.

**PROGRAM:**

import random

print(random.randint(0, 9))

**OUTPUT:**







**PROGRAM NO.: 4a**

**AIM:** Write a Python function that takes a list of words and returns the length of the longest one.

**THEORY:**

1. First declare a function with the name “longest Length” which accepts a list as an argument.
2. Now, take a list where you have all the values.
3. We are going to take this list, and we will iterate through each item using for loop.
4. Then we take two variables max1 and temp to store the maximum length and the word with the longest length.
5. After completing the above steps then we take the first value in the list and the first value length in order to compare.
6. Once the above steps are completed, we compare the items in the list using for loop. Below I had mentioned the logic.
7. After completing the above steps run the program, and then we’ll get the required result.

**PROGRAM:**

def longest\_string(strings):

longest = strings[0]

for i in strings:

if len(i) > len(longest):

longest = i

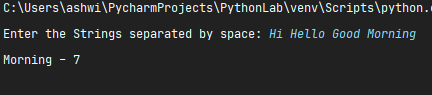
return longest + f' - {len(longest)}'

str\_list = input('Enter the Strings separated by space: ')

str\_list = str\_list.split(' ')

print(longest\_string(str\_list))

**OUTPUT:**



**PROGRAM NO.: 4b**

**AIM:** Write a Python program to count the number of characters (character frequency) in a string.

**THEORY:**

1. Define a string.
2. Define and initialize a variable count to 0.
3. Iterate through the string till the end and for each character except spaces, increment the count by 1.
4. First copy unique characters into empty list and then count frequency.

**PROGRAM:**

string = input('Enter the string to find the frequency: ')

string = string.lower()

unique\_characters = set(string)

occurrence = dict()

for i in unique\_characters:

occurrence[i] = 0

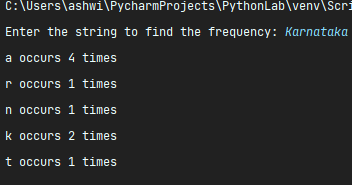
for i in string:

occurrence[i] += 1

for i in occurrence:

print(f'{i} occurs {occurrence[i]} times')

**OUTPUT:**



**PROGRAM NO.: 5a**

**AIM:** Finding the sum and average of given numbers using lists.

**THEORY:**

Iterate from (0 to len(list)+1). Sum=sum+i

Average=sum/(i+1)

**PROGRAM:**

my\_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]

my\_new\_list = []

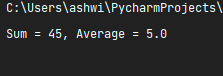
my\_sum = 0

for i in my\_list:

my\_sum = my\_sum + i

print(f'Sum = {my\_sum}, Average = {my\_sum / len(my\_list)}')

**OUTPUT:**



**PROGRAM NO.: 5b**

**AIM:** Write a Python program to remove duplicates from a list.

**THEORY:**

Copy unique elements from li1 into a new list.

**PROGRAM:**

original\_list = [1, 1, 2, 2, 3, 4, 5, 4, 6, 7]

new\_list = []

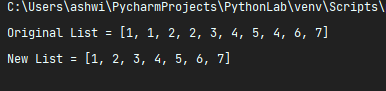
for i in original\_list:

if i not in new\_list:

new\_list.append(i)

print(f'Original List = {original\_list}\nNew List = {new\_list}')

**OUTPUT:**



**PROGRAM NO.: 5c**

**AIM:** Write a Python script to sort (ascending and descending) a dictionary by value.

**THEORY:**

Use inbuilt method sorted(d.items(),key=operator.itemgetter(1))

If we give reverse=True it prints in descending order

**PROGRAM:**

def dict\_sort():

d = {'ram': 5, 'ravi': 15, 'rahul': 10, 'Arun': 6}

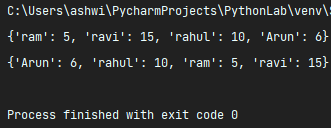
print(d)

dict1 = dict(sorted(d.items(), reverse=False))

print(dict1)

dict\_sort()

**OUTPUT:**



**PROGRAM NO.: 6a**

**AIM:** Write a Python program to add an item in a tuple.

**THEORY:**

Convert tuple into list, then append the new items into list. Now, convert the list back to tuple and print.

**PROGRAM:**

a = (1, 2, 3, 4, 5, 6)

print(f'The tuple is {a}')

a = list(a)

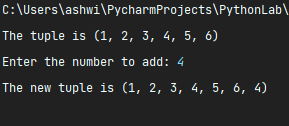
num = int(input('Enter the number to add: '))

a.append(num)

a = tuple(a)

print(f'The new tuple is {a}')

**OUTPUT:**



**PROGRAM NO.: 6b**

**AIM:** Write a Python program to add member(s) in a set.

**THEORY:**

Take the members as user input and use the “.add” method to add these members into set.

**PROGRAM:**

s = {1, 2, 3, 4, 5}

print(f'The set is {s}')

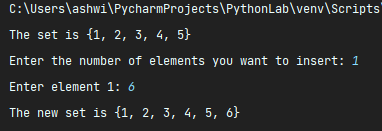
num = int(input('Enter the number of elements you want to insert: '))

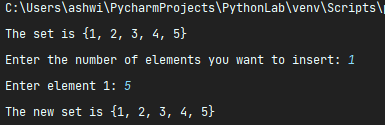
for i in range(num):

s.add(int(input(f'Enter element {i + 1}: ')))

print(f'The new set is {s}')

**OUTPUT:**





**PROGRAM NO.: 7a**

**AIM:** Write a Python function to find the Max of three numbers

**THEORY:**

Return the largest number among the 3 given numbers by using comparison operator.

**PROGRAM:**

def max\_num(a, b, c):

if a > b and a > c:

return a

elif b > c and b > a:

return b

else:

return c

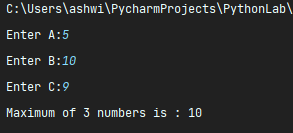
n1 = int(input('Enter A:'))

n2 = int(input('Enter B:'))

n3 = int(input('Enter C:'))

print(f'Maximum of 3 numbers is : {max\_num(n1, n2, n3)}')

**OUTPUT:**



**PROGRAM NO.: 7b**

**AIM:** Write a Python function to multiply all the numbers in a list

**THEORY:**

Multiply each element in the list using a function. For example, [1,2,3]=6

**PROGRAM:**

def list\_mul(nums):

product = 1

for i in nums:

product \*= i

return product

li = []

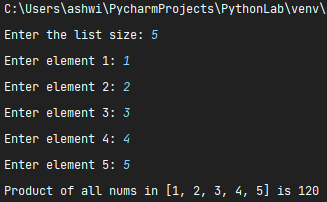
n = int(input('Enter the list size: '))

for i in range(n):

li.append(int(input(f'Enter element {i + 1}: ')))

print(f'Product of all nums in {li} is {list\_mul(li)}')

**OUTPUT:**



**PROGRAM NO.: 7c**

**AIM:** Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.

**THEORY:**

Here we use the concept of recursion to find the factorial of a number.

**PROGRAM:**

def fact(n):

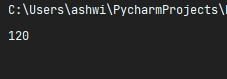
if n == 0 or n == 1:

return 1

return n \* fact(n - 1)

print(fact(5))

**OUTPUT:**



**PROGRAM NO.: 7d**

**AIM:** Write a Python program to sort a list of tuples using Lambda.

**THEORY:**

we take tuples containing name of subject and marks of students in a class. We shall sort these tuples based on the marks, the second element in all the tuples.

**PROGRAM:**

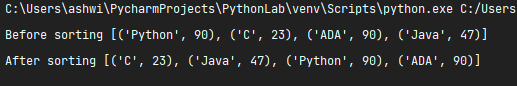
li1 = [('Python', 90), ('C', 23), ('ADA', 90), ('Java', 47)]

print(f'Before sorting {li1}')

li1.sort(key=lambda x: x[1])

print(f'After sorting {li1}')

**OUTPUT:**



**PROGRAM NO.: 8a**

**AIM:** Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

**THEORY:**

consider sum, x, y

x + y = sum

check this condition for every item in the list

**PROGRAM:**

class TwoNumberSum:

def method(self, array, targetSum):

pairs = []

for i in range(0, len(array)):

for j in range(0, len(array)):

if array[i] + array[j] is targetSum and array[i] != array[j]:

if (i, j) not in pairs and (j, i) not in pairs:

pairs.append((i, j))

return pairs

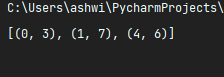
TNS = TwoNumberSum()

arr = [3, 5, 4, 8, 11, 1, 0, 6, 9]

s = 11

print(TNS.method(arr, s))

**OUTPUT:**



**PROGRAM NO.: 8b**

**AIM:** Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

**THEORY:**

Take length and breadth as user input.

Area=length\*breadth

**PROGRAM:**

class Rectangle:

def \_\_init\_\_(self, l, b):

self.l = l

self.b = b

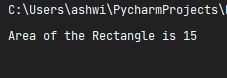
def area(self):

return self.l \* self.b

rec = Rectangle(3, 5)

print(f'Area of the Rectangle is {rec.area()}')

**OUTPUT:**



**PROGRAM NO.: 9a**

**AIM:** Write a Python GUI program to add a button in your application using tkinter module.

**THEORY:**

Initialize Tkinter

Initialize a button

Display the button in the main frame

**PROGRAM:**

import tkinter as tk

my\_window = tk.Tk()

button\_1 = tk.Button(my\_window, text='click\_me', height=1, width=35, command=my\_window.destroy)

button\_1.pack()

my\_window.mainloop()

**OUTPUT:**



**PROGRAM NO.: 9b**

**AIM:** Write a Python GUI program to create a Combo box with three options using tkinter module.

**THEORY:**

Initialize Tkinter

Initialize a Combo box

Display the Combo box in the main frame

**PROGRAM:**

import tkinter as tk

from tkinter import ttk

root = tk.Tk()

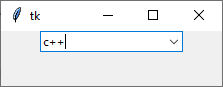
my\_str\_var = tk.StringVar()

my\_combobox = ttk.Combobox(root, textvariable=my\_str\_var, values=['c', 'c++', 'Java'])

my\_combobox.pack()

root.mainloop()

**OUTPUT:**



**PROGRAM NO.: 9c**

**AIM:** Write a Python GUI program to create a Check button widget using tkinter module.

**THEORY:**

Initialize Tkinter

Initialize a Check button

Display the Check button in the main frame

**PROGRAM:**

import tkinter as tk

from tkinter import ttk

root = tk.Tk()

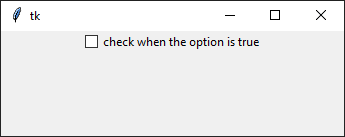
my\_boolean\_var = tk.BooleanVar()

my\_checkbutton = ttk.Checkbutton(text="check when the option is true", variable=my\_boolean\_var)

my\_checkbutton.pack()

root.mainloop()

**OUTPUT:**



**PROGRAM NO.: 10a**

**AIM:** Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text.

**THEORY:**

Initialize Tkinter

Initialize a Text widget

Display the Text widget in the main frame

**PROGRAM:**

import tkinter as tk

parent = tk.Tk()

mytext = tk.Text(parent)

mytext.insert('1.0', 'i like python')

mytext.insert('1.2', 'dont ')

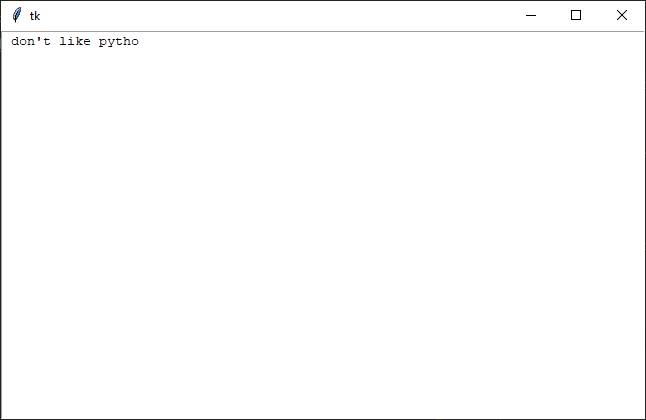
mytext.delete('1.0')

mytext.delete('end-2 chars')

mytext.pack()

parent.mainloop()

**OUTPUT:**



**PROGRAM NO.: 10b**

**AIM:** Write a Python GUI program to create three radio buttons widgets using tkinter module.

**THEORY:**

Initialize Tkinter

Initialize three radio buttons

Display the radio buttons in the main frame

**PROGRAM:**

import tkinter as tk

parent = tk.Tk()

parent.title("Radiobutton")

parent.geometry("250x50")

radio1 = tk.Radiobutton(parent, text='first', value=1)

radio2 = tk.Radiobutton(parent, text='second', value=2)

radio3 = tk.Radiobutton(parent, text='third', value=3)

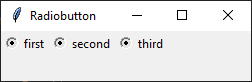
radio1.grid(column=0, row=0)

radio2.grid(column=1, row=0)

radio3.grid(column=2, row=0)

parent.mainloop()

**OUTPUT:**



**PROGRAM NO.: 10c**

**AIM:** Write a Python GUI program to create a Listbox bar widgets using tkinter module.

**THEORY:**

Initialize Tkinter

Initialize three radio buttons

Display the radio buttons in the main frame

**PROGRAM:**

import tkinter as tk

parent = tk.Tk()

parent.geometry("250x200")

label1 = tk.Label(parent, text='a list of favourite languages...')

listbox = tk.Listbox(parent)

listbox.insert(1, 'PHP')

listbox.insert(2, 'pyhton')

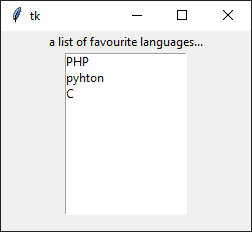
listbox.insert(3, 'C')

label1.pack()

listbox.pack()

parent.mainloop()

**OUTPUT:**



**PROGRAM NO.: 11**

**AIM:** Write a Python program to create a small GUI application for insert, update and delete in a table using a backend and front end for creating form.

**THEORY:**

Connection to SQL

Create Table -> Insert Data -> Update Data -> Delete Data -> Drop Table

**PROGRAM:**

from tkinter import \*

import mysql.connector

# create root window

root = Tk()

root.title('Database with GUI')

root.config(background='red')

# create table inside database

db = mysql.connector.connect(host='localhost', user='root', passwd='1379', database='lab')

cursor = db.cursor()

cursor.execute('create table if not exists class(Name varchar(20),usn int(20))')

db.commit()

textin = StringVar()

textinn = StringVar()

def insert():

name1 = textin.get()

usn1 = textinn.get()

cursor.execute("INSERT INTO class (Name, usn) VALUES (%s, %s)", (name1, usn1))

db.commit()

def show():

cursor.execute("select \* from class")

for row in cursor.fetchall():

print(row)

lab = Label(root, text="Name")

lab.place(x=0, y=0)

entname = Entry(root, width=20, textvar=textin)

entname.place(x=80, y=0)

lab1 = Label(root, text="usn")

lab1.place(x=0, y=40)

entusn = Entry(root, width=20, textvar=textinn)

entusn.place(x=80, y=40)

but = Button(root, padx=2, pady=2, text="submit", command=insert)

but.place(x=60, y=100)

res = Button(root, padx=2, pady=2, text="show", command=show)

res.place(x=160, y=100)

del1 = StringVar()

def delt():

delv = del1.get()

cursor.execute('DELETE FROM class WHERE Name = %s', (delv,))

db.commit()

labdelete = Label(root, text="delete")

labdelete.place(x=0, y=340)

endelete = Entry(root, width=20, textvar=del1)

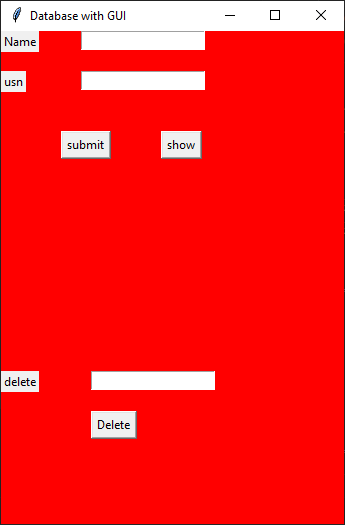
endelete.place(x=90, y=340)

butdel = Button(root, padx=2, pady=2, text="Delete", command=delt)

butdel.place(x=90, y=380)

root.mainloop()

**OUTPUT:**



**PROGRAM NO.: 12a**

**AIM:** Write a Python program to add, subtract, multiple and divide two Pandas Series.

**THEORY:**

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,

built on top of the Python programming language.

**PROGRAM:**

import pandas as pd

ds1 = pd.Series([2, 4, 6, 8])

ds2 = pd.Series([1, 3, 5, 7])

dsum = ds1 + ds2

dsub = ds1 - ds2

dmul = ds1 \* ds2

ddiv = ds1 / ds2

print(ds1)

print(ds2)

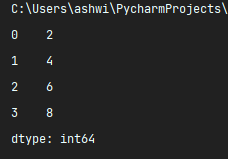
print(dsum)

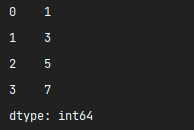
print(dsub)

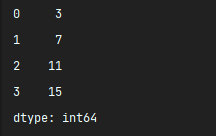
print(dmul)

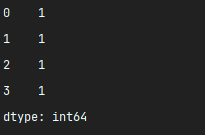
print(ddiv)

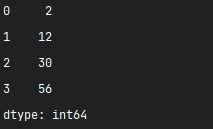
**OUTPUT:**

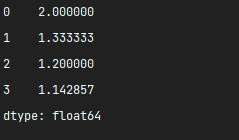












**PROGRAM NO.: 12b**

**AIM:** Write a Python program to convert a dictionary to a Pandas series.

**THEORY:**

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,

built on top of the Python programming language.

**PROGRAM:**

import pandas as pd

d1 = {'a': 100, 'b': 200, 'c': 300, 'd': 400}

print('Dictionary', d1)

nd = pd.Series(d1)

print(f'Pandas Series\n{nd}')

**OUTPUT:**

