**PYTHON PRACTICAL FILE**

**NAME:** Vaishali Mishra

**COURSE:** B.Sc. (Hons.) Computer Science

**YEAR/SEMESTER:** 2nd Year, 3rd Sem

**SUBMITTED TO:**  Subodh Sir

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **PRACTICAL QUESTION** | **PAGE NO.** |
| **1** | Write a function that takes the lengths of three sides : side1, side2 & side3 of the triangle as the input from the user using input function and return the area and perimeter of the triangle as a tuple. Also, assert that sum of the length of any two sides is greater than the third side. | **3-4** |
| **2** | Consider a show room of electronic products ,where there are various  salesmen. Each sales man is given acommission of 5% , depending on the sales made per month. Incase the sale done is less than 50000 ,then the sales man is not given any commission. Write a function to calculate total sales of a salesman in a month ,commission and remarks for the salesman.Sales done by each sales man per week is to be provided as input. Use tuples/list to store data of salesmen.  Assign remarks according to the following criteria:  Excellent:Sales>=80000  Good:Sales>=60000 and<80000  Average:Sales>=40000 and<60000  WorkHard:Sales<40000 | **4-5** |
| **3** | Write a Python function to find the nth term of Fibonacci sequence and its factorial. Return the result as a list. | **5-6** |
| **4** | Write a function that takes a number (>=10) as an input and return the digits of the number as a set. | **6** |
| **5** | Write a function that finds the sum of the n terms of the following series. Import the factorial function created in question 3.  1–x2/2!+x4/4!–x6/6!+…xn/n! | **6-7** |
| **6** | Consider a tuple t1={1,2,5,7,9,2,4,6,8,10}. Write a program to perform following operations:  a)Print another tuple whose values are even numbers in the given tuple.  b)Concatenate a tuple t2={11,13,15) with t1.  c)Return maximum and minimum value from this tuple. | **7-8** |
| **7** | Write a menu driven program to perform the following on strings:  a)Find the length of string.  b)Return maximum of three strings.  c)Accept a string and replace all vowels with“#”  d)Find number of words in the given string.  e)Check whether the string is a palindrome or not | **8-13** |

|  |  |  |
| --- | --- | --- |
| **8** | Write a Python program to perform the following using list:  a)Check if all elements in list are numbers or not.  b)If it is a numeric list, then count number of odd values in it.  c)If list contains all Strings ,then display largest String in the list.  d)Display list in reverse form.  e)Find a specified element in list.  f)Remove the specified element from the list.  g)Sort the list in descending order.  h)accept 2 lists and find the common members in them. | **14-21** |
| **9** | Use dictionary to store marks of the students in 4 subjects. Write a function to Find the name of the student securing highest percentage.(Hint: Names of students are unique). | **22-23** |
| **10** | Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count. | **23-24** |
| **11** | Write a menu-driven program to accept a list of student names and perform the following   1. search an element using linear search / binary search. 2. Sort the elements using bubble sort /insertion sort/ selection sort. | **24-28** |
| **12** | Write a program that makes use of a function to accept a  list of n integers. And displays a histogram | **29-30** |
| **13** | Write a program that makes use of a function to display sine, cosine, Polynomial and exponential curves. | **30-32** |
| **14** | Write a function that reads a file file1 and copies only alternative lines to Another file file2. Alternative lines copied should be the odd numbered lines. Use Exception. | **32-33** |
| **15** | Define a class Student to store his/her name and marks in three subjects. Use a class variable to store the maximum average marks of the class. Use Constructor and destructor to initialize and destroy the objects. | **33-35** |

# #Ques 1

**#Write a function that takes the lengths of three sides:side1, side2 & side3**

**#of the triangle as the input from the user using input function and return**

**#the area and perimeter of the triangle as a tuple. Also, assert that sum of**

**#the length of any two sides is greater than the third side.**

import math

def triangle():

side1=int(input("Enter side1 - ")) side2=int(input("Enter side2 - ")) side3=int(input("Enter side3 - "))

perimeter=0 area=0 t=(perimeter,area)

if((side1+side2)>side3 and (side2+side3)>side1 and (side1+side3)>side2): print("Two sides are greater than the third")

else:

print("Two sides are not greater than the third")

# Check if input forms a triangle

if side1 + side2 > side3 and side2 + side3 > side1 and side1 + side3 >

side2: print("Given sides form a triangle.") perimeter = side1 + side2 + side3 s = perimeter/2

area = math.sqrt(s\*(s-side1)\*(s-side2)\*(s-side3))

else:

print("Error: Given sides does not form a triangle.") print("[Sum of two sides of a triangle should be greater than the

third]") return perimeter, area

t1=triangle() print("Perimeter =",t1[0]) print("Area =",t1[1])

# #Ques 2

**#Consider a showroom of electronic products, where there are various salesmen.**

**#Each salesman is given a commission of 5%, depending on the sales made per**

**#month. In case the sale done is less than 50000, then the salesman is not given**

**#any commission. Write a function to calculate total sales of a salesman in a**

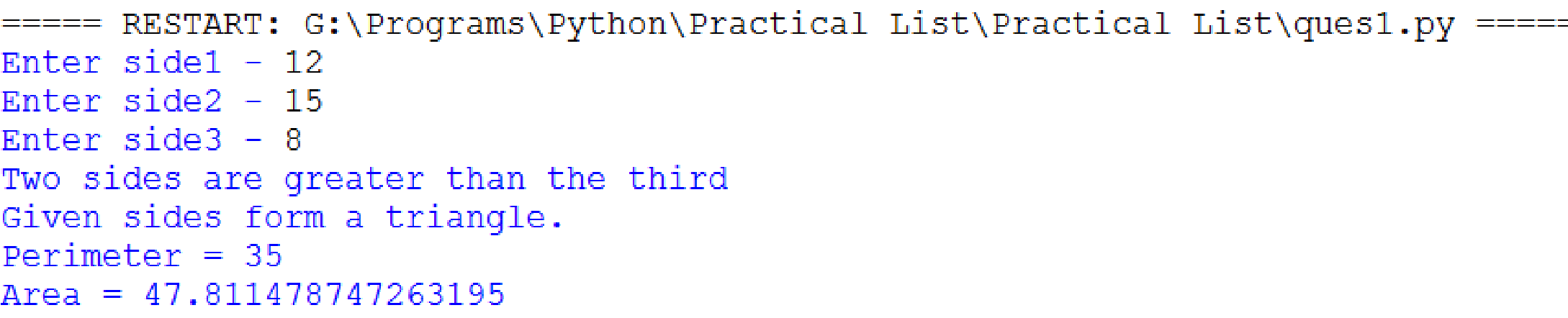
**#month, commission and remarks for the salesman. Sales done by each sales man**

**#per week is to be provided as input. Use tuples/list to store data of salesmen.**

**#Assign remarks according to the following criteria:**

**#Excellent : Sales>=80000**

**#Good : Sales>=60000 and <80000**

**#Average : Sales>=40000 and <60000 #Work Hard : Sales<40000**

def accounts(): name=input("Enter the name of the salesman : ") sales=0 sales+=int(input("Enter the sales in week 1 : ")) sales+=int(input("Enter the sales in week 2 : ")) sales+=int(input("Enter the sales in week 3: ")) sales+=int(input("Enter the sales in week 4 : "))

commission=0.0;remarks= " " if sales>=50000: commission=sales\*0.05

if sales>=80000:

remarks="Excellent"

elif sales>=60000 and sales<80000: remarks="Good"

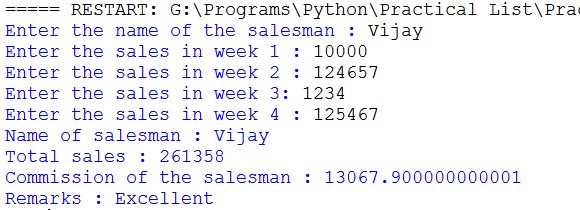
elif sales>=40000 and sales<60000: remarks="Average" elif sales<4000:

remarks="Work Hard"

tri=(name,sales,commission,remarks) return tri

tup=accounts() print("Name of salesman :",tup[0]) print("Total sales :",tup[1])

print("Commission of the salesman :",tup[2]) print("Remarks :",tup[3])



# #Ques 3

**#Write a Python function to find the nth term of Fibonacci sequence and its**

**#factorial. Return the result as a list.**

# Factorial function to find factorial of a number def factorial(n):

fact=1 for i in range(1,n+1): fact\*=i

return fact

#Recursive Fibonacci function to find nth term of series def fibonacci(n): if n < 0:

return 0

if n == 1 or n==0:

return n

return fibonacci(n-1) + fibonacci(n-2)

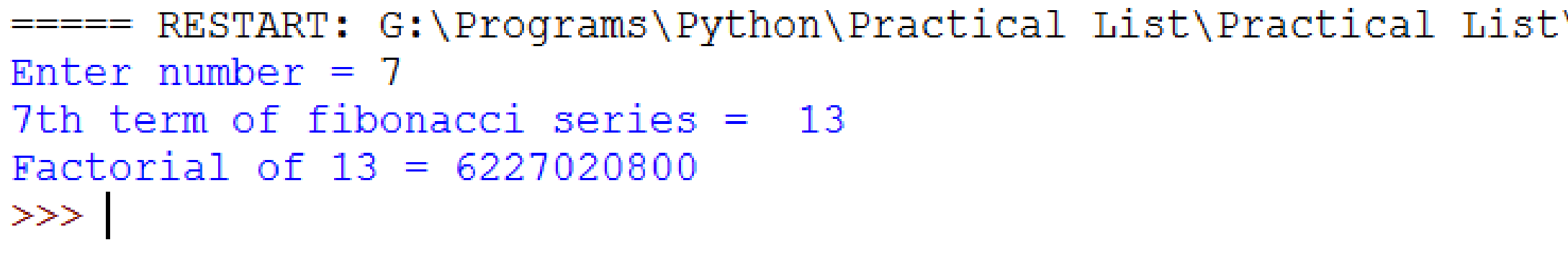
# Main if \_\_name\_\_ == "\_\_main\_\_": num=int(input("Enter number = ")) fib = fibonacci(num) facto = factorial(fib)

print(str(num)+"th term of fibonacci series = ",fib) print("Factorial of "+str(fib)+" =",facto)

# #Ques 4

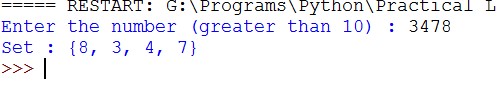
**#Write a function that takes a number(>=10) as an input and return the #digits of the number as a set.**

def digits(x): digits\_num=set() while(x!=0): digits\_num.add(x%10) x=x//10

return digits\_num

num=int(input("Enter the number (greater than 10) : ")) if num<10: print("Number less than 10")

else: print("Set :",digits(num))



# #Ques 5

**#Write a function that finds the sum of the n terms of the following series.**

**#Import the factorial function created in question4.**

**#1–x2/2!+x4/4!–x6/6!+…xn/n!**

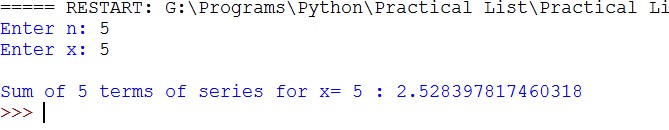
# Importing factorial function created previously from ques3 import factorial

# Function to find sum of given series def sum\_series(x, n): sum = 0 for i in range(1, n+1, 1): term = ((-1)\*\*(i+1))\*((x\*\*(2\*i-2)/factorial(2\*i-2))) #Calculating nth term sum += term

return sum

if \_\_name\_\_ == "\_\_main\_\_": n = int(input("Enter n: ")) x = int(input("Enter x: ")) sum = sum\_series(x, n)

print("\nSum of",n,"terms of series for x=",x,":",sum)



# #Ques 6

**#Consider a tuple t1={1,2,5,7,9,2,4,6,8,10}. Write a program to perform #following operations:**

**#a) Print another tuple whose values are even numbers in the given tuple.**

**#b) Concatenate a tuple t2={11,13,15) with t1.**

**#c) Return maximum and minimum value from this tuple.**

#Tuple t1 t1=(1,2,5,7,9,2,4,6,8,10) print("Given tuple -",t1)

#Another tuple with even values of t1 t1\_even=() for x in t1:

if x%2==0:

t1\_even+=(x,)

print("New tuple with even values -",t1\_even)

# Concatenate t2 with t1 t2 = (11, 13, 15) t3 = t1 + t2 print("Concatented tuple: ", t3)

# Maximum and minimum values from concatenated tuple maximum = t3[0] minimum = t3[0] for i in range(0, len(t3), 1):

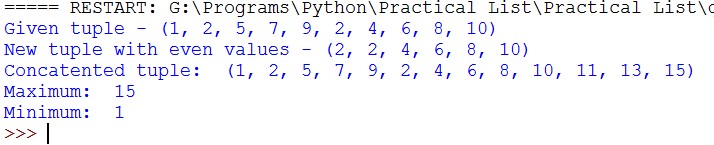
if t3[i] > maximum:

maximum = t3[i]

if t3[i] < minimum:

minimum = t3[i]

print("Maximum: ", maximum) print("Minimum: ", minimum)



# #Ques 7

**#Write a menu driven program to perform the following on strings:**

**#a) Find the length of string.**

**#b) Return maximum of three strings.**

**#c) Accept a string and replace all vowels with "#" #d) Find number of words in the given string. #e) Check whether the string is a palindrome or not.**

def len\_str(): str = input("Enter the string: ")

count=0 for i in str:

count+=1

print("Length of string:",count)

def maxof\_three():

str1 = input("Enter string 1: ") str2 = input("Enter string 2: ") str3 = input("Enter string 3: ") maxstr = "" if str1 > str2 and str1 > str3: maxstr = str1

elif str2 > str1 and str2 > str3:

maxstr = str2

else:

maxstr = str3

print("Maximum of above three: ", maxstr)

def replace\_vowels():

str = input("Enter the string: ") new\_str = ""

vowels = ['a','e','i','o','u'] for i in range(0, len(str)):

if str[i].lower() in vowels:

new\_str += "#"

else:

new\_str += str[i]

print("Replaced string: ", new\_str)

def numofwords():

str = input("Enter the string: ")

str = str.strip() + " " count = 0 for i in range(0, len(str)):

if str[i] == " ": count += 1

print("No of words: ", count)

def palindrome(): str = input("Enter the string: ") new\_str = "" for i in range(0, len(str)):

new\_str = str[i] + new\_str

if str == new\_str: print(str,"is a palindrome.")

else: print(str,"is not a palindrome")

if \_\_name\_\_ == "\_\_main\_\_": ch = 'y' while ch.lower() == 'y': print("\nMenu") print("1. Length of string") print("2. Maximum of three strings") print("3. Replace vowels with '#'") print("4. No. of words") print("5. Check Palindrome") print("6. Exit") option = input("Your choice: ") if(option=='1'):

len\_str()

elif(option=='2'):

maxof\_three() elif(option=='3'):

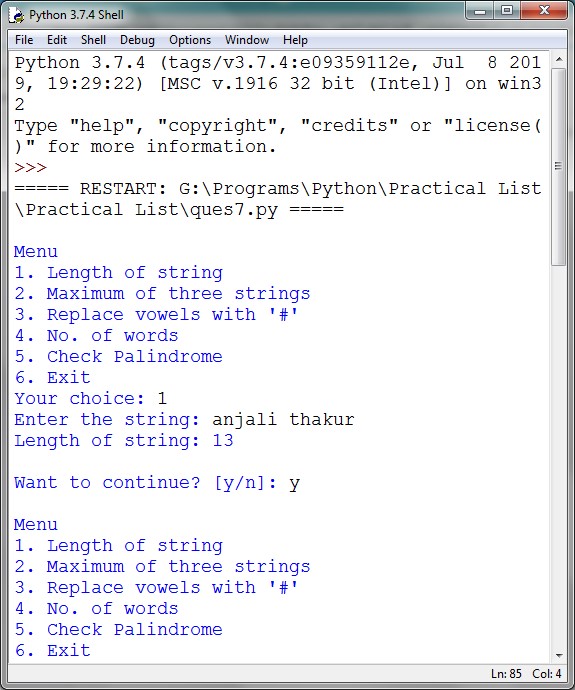
replace\_vowels()

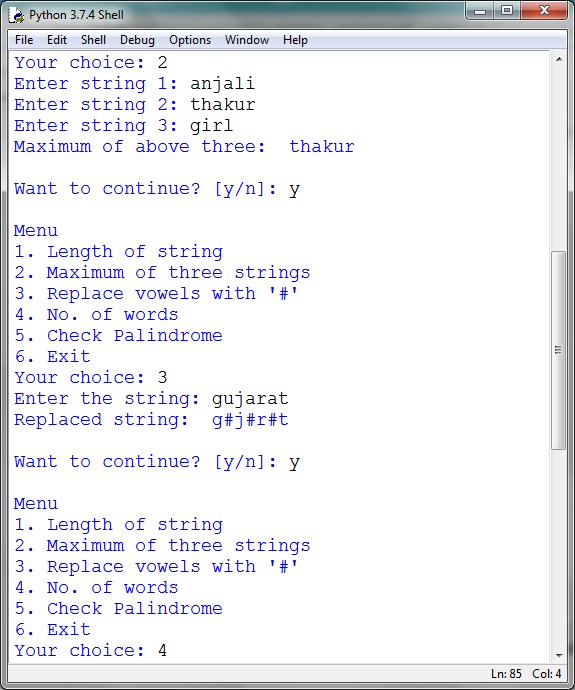
elif(option=='4'): numofwords() elif(option=='5'):

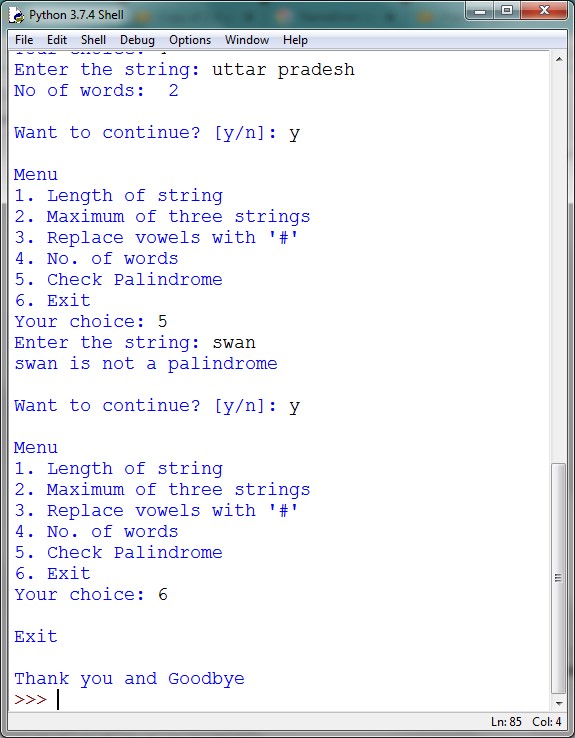
palindrome()

elif(option=='6'): print("\nExit") print("\nThank you and Goodbye") break

else: print("Choose a valid option") ch = input("\nWant to continue? [y/n]: ")







# #Ques 8

**#Write a Python program to perform the following using list:**

**#a) Check if all elements in list are numbers or not.**

**#b) If it is a numeric list, then count number of odd values in it.**

**#c) If list contains all Strings, then display largest String in the list.**

**#d) Display list in reverse form.**

**#e) Find a specified element in list.**

**#f) Remove the specified element from the list.**

**#g) Sort the list in descending order.**

**#h) Accept 2 lists and find the common members in them.**

# Check if all elements of list are numbers or not def check\_int(l):

for i in range(0, len(l),1):

if type(l[i]) != int:

return False

return True

# If it is a numeric list, then count number of odd values def count\_odd(l):

if check\_int(l): count = 0 for i in range(0,len(l),1):

if int(l[i]) % 2 != 0:

count += 1

print("Count of odd numbers: ", count) return

print("Not all are numbers")

# If list contains all strings, then display largest string in the list def largest\_str(l): flag = True for i in range(0, len(l), 1):

if type(l[i]) != str:

flag = False

if flag: largest = l[0] for i in l:

if len(i) > len(largest): largest = i

print("Largest string: ", largest)

else: print("List does not contain all strings!")

# Display list in reverse form def display\_reverse(l):

for i in range(len(l)-1, -1, -1):

print(l[i], end=" ")

return

# Find a specific item in the list def find\_item(l): item = input("\nEnter an element: ") for i in range(0, len(l), 1):

if item == l[i]: print("Item found at index: ", i)

return

print("Item not found")

# Remove the specified item in the list def remove\_item(l): item = eval(input("\nEnter an element: ")) for i in range(0, len(l), 1):

if item == l[i]:

l.remove(item) print("Item removed") return

print("Item couldn't be removed") return

# Sort the list in descending order def sort\_desc(l):

for i in range(len(l)):

max\_idx = i for j in range(i+1, len(l)):

if l[max\_idx] < l[j]:

max\_idx = j

l[i], l[max\_idx] = l[max\_idx], l[i]

print("Sorted list in descending order -",l) return

# Accept two list and find common members in them def common(l1, l2): common = [] for i in range(0, len(l1), 1):

for j in range(0, len(l2), 1): if l1[i] == l2[j]:

common.append(l1[i])

if common: print("Common elements: ", common) else:

print("No common element")

return

if \_\_name\_\_ == "\_\_main\_\_": l = [] n = int(input("Enter the size of list: ")) for i in range(0, n,1):

l.append(eval(input("Enter element"+str(i+1)+": ")))

ch = 'y' while ch.lower() == 'y':

print("\nMenu") print("1. Check if all elements are numbers") print("2. Count odd numbers if list is numeric") print("3. Display largest string in list") print("4. Reverse the list") print("5. Find item in list") print("6. Remove item from list") print("7. Sort in Descending order") print("8. Find common elements from another list") print("9. Exit") option = input("Your choice: ")

if(option=='1'):

if check\_int(l): print("All elements are numbers")

else: print("All elements are not numbers")

elif(option=='2'):

count\_odd(l)

elif(option=='3'):

largest\_str(l)

elif(option=='4'):

display\_reverse(l)

elif(option=='5'):

find\_item(l)

elif(option=='6'):

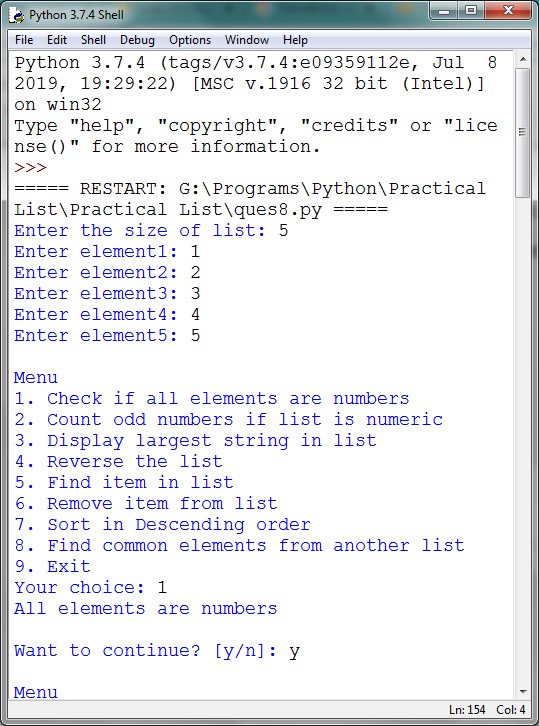
remove\_item(l) elif(option=='7'):

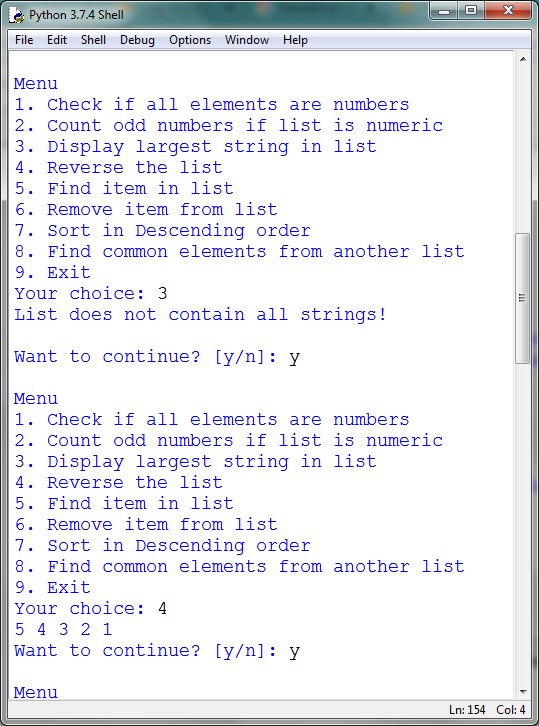
sort\_desc(l)

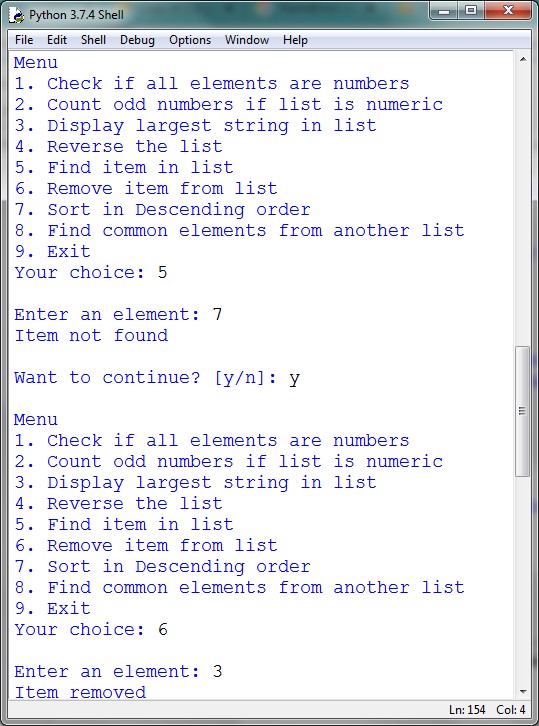
elif(option=='8'): l2 = [] n = int(input("Enter the size of new list: ")) for i in range(0, n, 1): l2.append(eval(input("Enter element"+str(i+1)+": "))) common(l, l2)

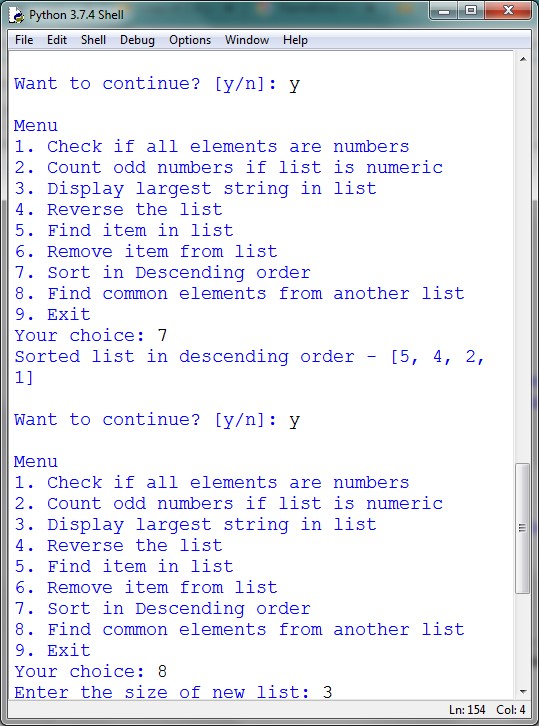
elif(option=='9'): print("\nExit") print("\nThank you and Goodbye") break

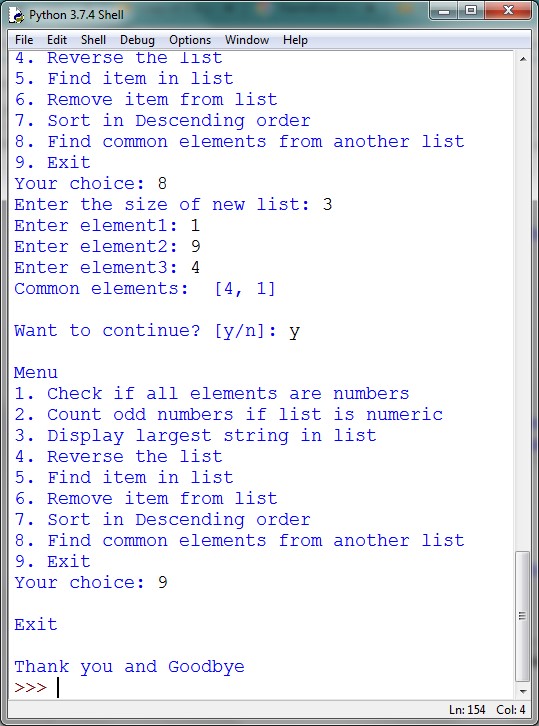
else: print("Choose a valid option") ch = input("\nWant to continue? [y/n]: ")











# #Ques 9

**#Use dictionary to store marks of the students in 4 subjects. #Write a function to find the name of the student securing #highest percentage. (Hint: Names of students are unique).**

def findTopStudent(marks):

name = '' percentage = 0.0 for student in marks: marks\_list = marks[student] total = 0.0 for i in range(0, len(marks\_list)): total += marks\_list[i]

if percentage < total/4: percentage = total/4 name = student

return name

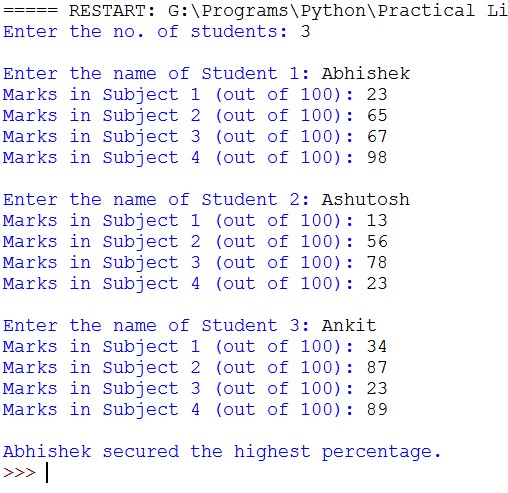
if \_\_name\_\_ == "\_\_main\_\_": marks = {} num = int(input('Enter the no. of students: ')) for i in range(1, num+1): name = input('\nEnter the name of Student '+str(i)+': ') temp = [] for j in range(1, 5): mark = float(input('Marks in Subject '+str(j)+' (out of 100): '))

temp.append(mark)

marks[name] = temp

topper = findTopStudent(marks)

print('\n'+topper+' secured the highest percentage.')



# #Ques 10

**#Write a function that takes a sentence as input from the user #and calculates the frequency of each letter.**

**#Use a variable of dictionary type to maintain the count.**

def inp(): str1=input("Enter the sentence : ")

dict = {} for n in str1:

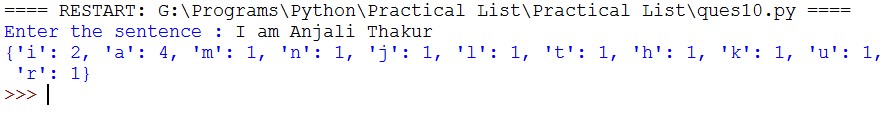
letter = n.lower() if letter not in [' ', '.', ',', '\'', '\"', '!', ';']: if letter in dict:

dict[letter] += 1

else:

dict[letter] = 1

print(dict) inp()



# #Ques 11

**#Write a menu-driven program to accept a list of student names and**

**#perform the following**

**#a) search an element using linear search/ binary search.**

**#b) Sort the elements using bubble sort/ insertion sort/ selection sort.**

def bubbleSort(list):

for i in range(0, len(list)-1, 1):

for j in range(0, len(list) - i - 1, 1):

if list[j] > list[j+1]: temp = list[j] list[j] = list[j+1] list[j+1] = temp

return list

def insertionSort(list):

for i in range(0, len(list), 1): temp = list[i] j = i - 1 while j >= 0 and list[j] > temp:

list[j+1] = list[j] j -= 1

list[j+1] = temp

return list

def selectionSort(list):

for i in range(0, len(list)-1, 1): minIndex = i for j in range(i+1, len(list), 1):

if list[minIndex] > list[j]:

minIndex = j

temp = list[minIndex] list[minIndex] = list[i] list[i] = temp return list

def linearSearch(list, element):

for i in range(0, len(list), 1):

if list[i].lower() == element.lower(): return i

return -1

def binarySearch(list, element):

list = selectionSort(list) low = 0 high = len(names) - 1 while low <= high:

mid = int(low + (high - low) / 2) if (list[mid] == element): return mid

if (list[mid] > element): high = mid - 1

else:

low = mid + 1

return -1

if \_\_name\_\_ == "\_\_main\_\_": num = int(input('\nEnter the number of students: ')) print('Enter the names of students:')

names = [] for i in range(0, num, 1): names.append(input(str(i+1)+':'))

choice = 'y' while choice.lower() == 'y': print('\n-------- Menu --------') print('1. Search a name') print('2. Sort the list of names') choice = input('Your Choice: ') if choice == '1':

name = input('\nEnter a name to search: ')

choice = input('Choose a searching algorithm:\n1. Linear, 2. Binary: ') index = -1 if choice == '1': index = linearSearch(names, name)

elif choice == '2':

index = binarySearch(names, name)

else: print('Invalid Choice!')

if index == -1: print('Name is not in the list.') else:

print('Name found in the list.')

elif choice == '2': choice = input('Choose a sorting algorithm:\n1. Bubble, 2. Insertion,

3. Selection: ') sorted\_names = [] if choice == '1': sorted\_names = bubbleSort(names) elif choice == '2': sorted\_names = insertionSort(names)

elif choice == '3':

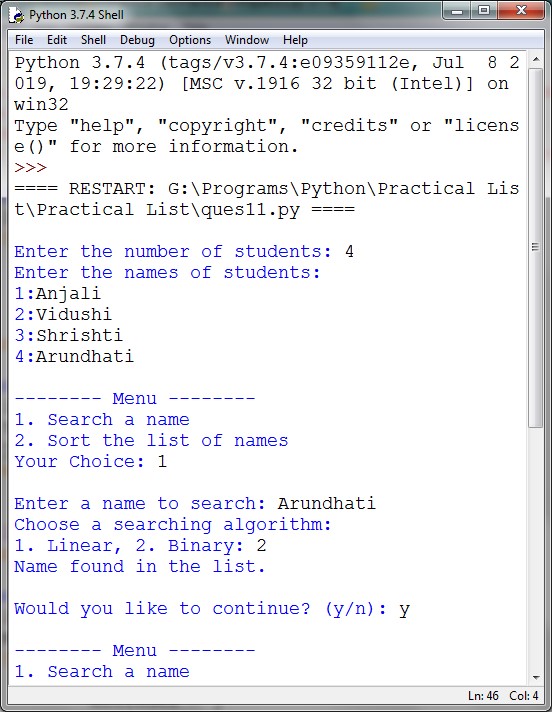
sorted\_names = selectionSort(names)

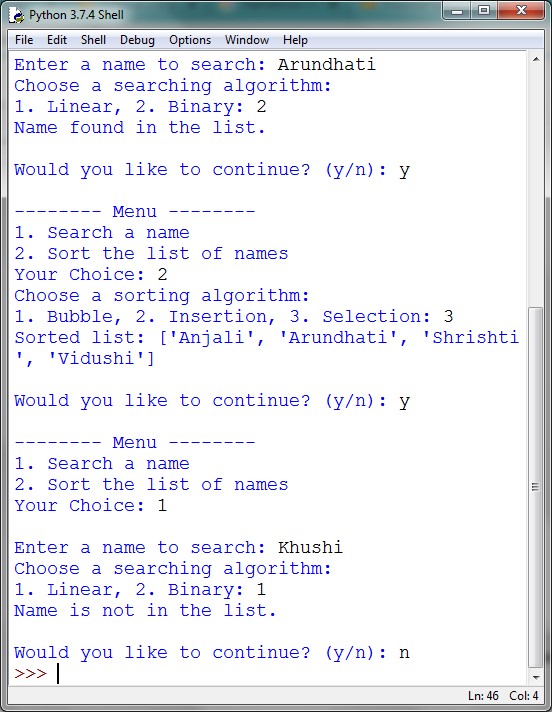
else:

print('Invalid Choice!')

print('Sorted list: ', end='') print(sorted\_names)

else: print('Invalid Choice!') choice = input('\nWould you like to continue? (y/n): ')





# #Ques 12

**#Write a program that makes use of a function to accept a list of #n integers and displays a histogram.**

import matplotlib.pyplot as plt

def inputList():

#To take a input of list of integers from user

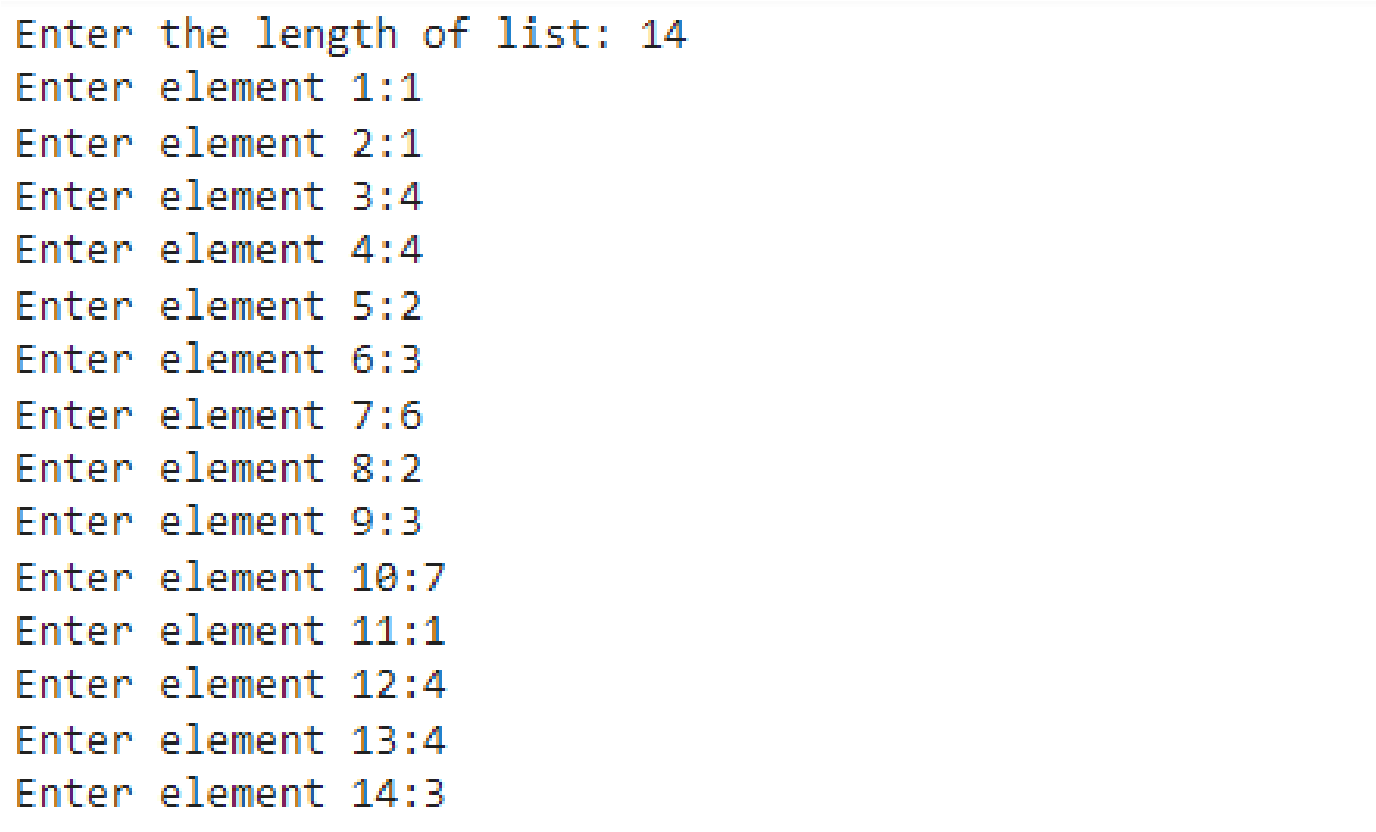
#Returns the list

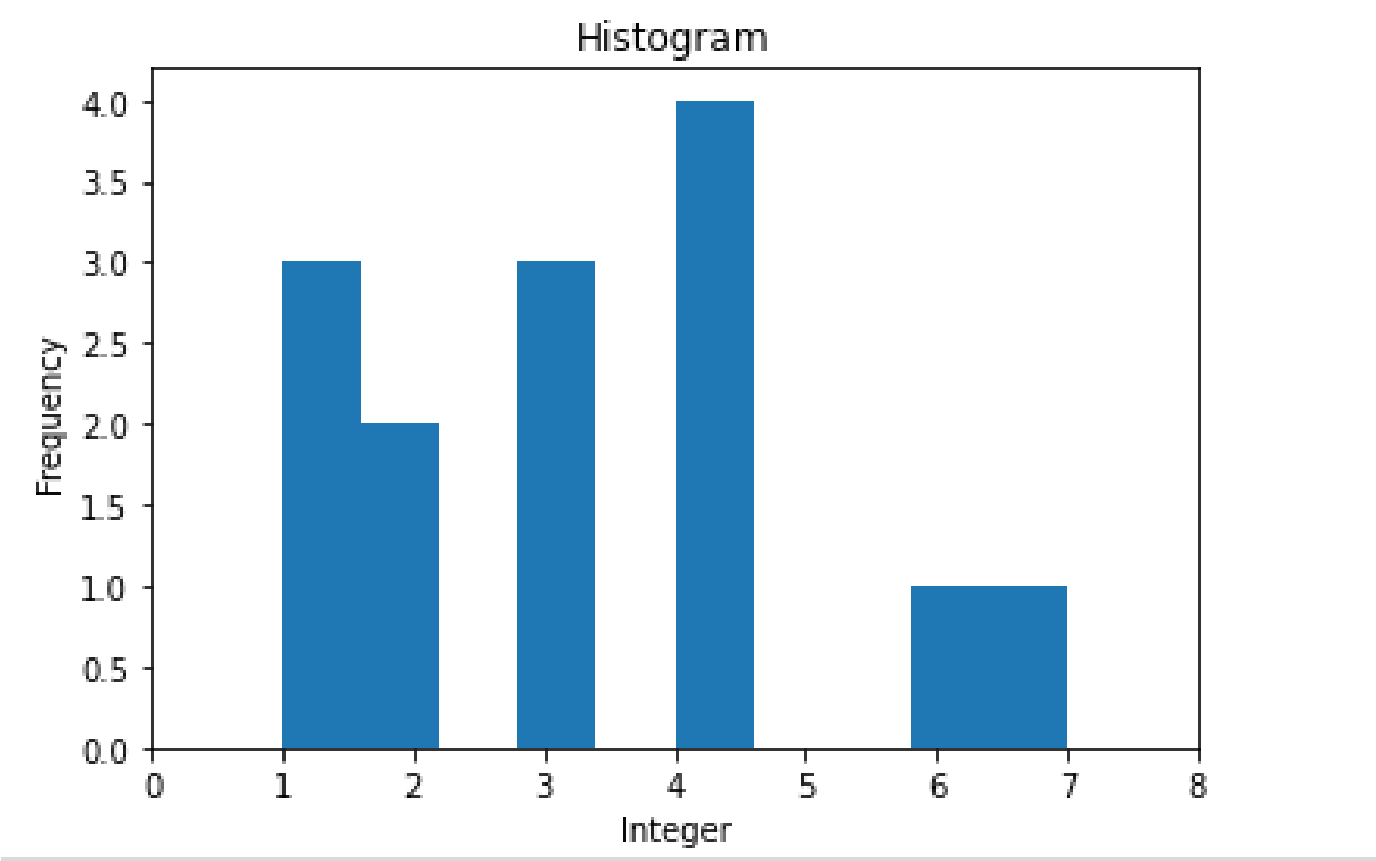
ls = [] length = int(input('Enter the length of list: ')) for i in range(0, length, 1):

ls.append(int(input('Enter element '+str(i+1)+':')))

return ls if \_\_name\_\_ == "\_\_main\_\_":

list = inputList() plt.hist(list) plt.xlabel('Integer') plt.ylabel('Frequency') plt.title('Histogram') plt.xlim(min(list)-1 , max (list)+1) #defining limits plt.show()





# #Ques 13

**#Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.**

from matplotlib import pyplot as plt import math

def sineCurve(): #To plot sine function plt.subplot(2, 2, 1) degrees = range(0, 360 + 1) sinValues = [math.sin(math.radians(i)) for i in degrees] plt.plot(sinValues) plt.xlabel('Degrees') plt.ylabel('Sin Values') plt.title('Sine Curve') plt.grid()

def cosineCurve(): #To plot cos function plt.subplot(2, 2, 3) degrees = range(0, 360 + 1) cosValues = [math.cos(math.radians(i)) for i in degrees] plt.plot(degrees, cosValues) plt.xlabel('Degrees') plt.ylabel('Cosine Values')

plt.title('Cosine Curve') plt.grid()

def polynomialCurve(): #To plot a polynomial function def polynomial(x):

return (8\*x\*x)

plt.subplot(2, 2, 2) x = range(-51, 50 + 2) y = [polynomial(i) for i in x] plt.plot(x, y) plt.xlabel('x') plt.ylabel('y = 8x^2') plt.title('Polynomial Curve') plt.grid()

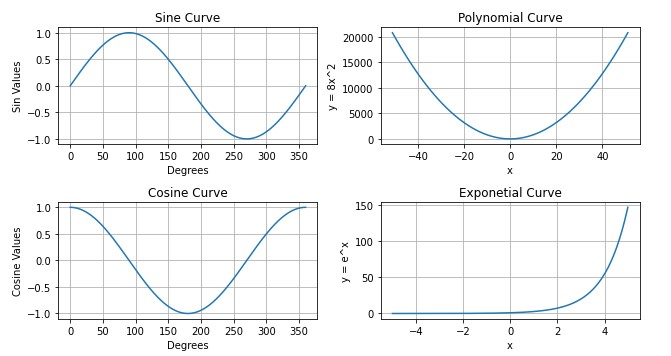
def expCurve(): #To plot exponential curve plt.subplot(2, 2, 4) x = [] for i in range(0, 100\*10):

x.append((-5) + (0.01)\*i)

y = [math.exp(i) for i in x] plt.plot(x, y) plt.xlabel('x') plt.ylabel('y = e^x') plt.title('Exponetial Curve') plt.grid()

if \_\_name\_\_ == "\_\_main\_\_":

plt.figure(figsize=(9, 5)) # To set the figure size sineCurve() cosineCurve() polynomialCurve() expCurve() plt.tight\_layout() plt.show()



# #Ques 14

**#Write a function that reads a file file1 and copies only alternative lines**

**#to another file file2. Alternative lines copied should be the odd numbered**

**#lines. Use Exception.**

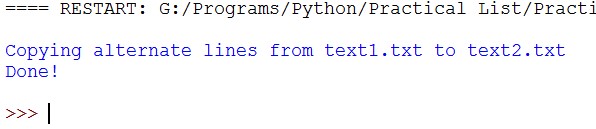
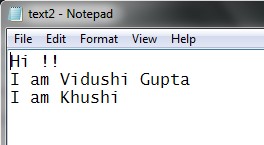
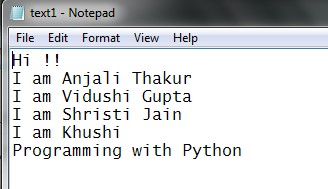
def copyOddNumberedLines(file\_in, file\_out): print("\nCopying alternate lines from "+file\_in+" to "+file\_out) try: fh\_in = open(file\_in, 'r') fh\_out = open(file\_out, 'w') lines = fh\_in.readlines() for i in range(0, len(lines), 2):

fh\_out.write(lines[i])

print('Done!\n')

except FileNotFoundError: print('Error: '+file\_in+' not found.\n') if \_\_name\_\_ == "\_\_main\_\_":

copyOddNumberedLines('text1.txt', 'text2.txt')



# #Ques15

**#Define a class Student to store his/ her name and marks in three subjects.**

**#Use a class variable to store the maximum average marks of the class. #Use constructor and destructor to initialize and destroy the objects.**

class Student: # Class Variable max\_avg = 0

# Constructor def \_\_init\_\_(self, name='', marks=[0, 0, 0]):

self.name = name self.marks = marks

# Destructor def \_\_del\_\_(self): del self.name del self.marks del self

def display(self): print("\nMarks: ",self.name) print("------------------") print("Subject 1:",self.marks[0]) print("Subject 2:",self.marks[1]) print("Subject 3:",self.marks[2]) print("Average:",self.calcAvg())

def calcAvg(self): total = 0.0 for i in range(0, 3, 1):

total += self.marks[i]

return total/3 if \_\_name\_\_ == "\_\_main\_\_":

students = [] num = int(input('\nEnter the no. of students: ')) for i in range(1, num+1, 1): name = input('\nEnter the name of Student '+str(i)+': ') marks = [] for j in range(1, 4, 1): marks.append(float(input('Marks in Subject '+str(j)+': ')))

student = Student(name, marks) if Student.max\_avg < student.calcAvg():

Student.max\_avg = student.calcAvg() students.append(student)

for i in range(0, len(students), 1):

print(students[i].display()) print('\nMaximum average marks of class: ',Student.max\_avg)

