**PSG College of Technology, Coimbatore-04**

**Department of Applied mathematics and Computational Sciences**

**20XC28 – Python Programming Lab**

**Problem Sheet – IV – Lists (One dimensional)**

1. Write a program that reads integers from the user and stores them in a list. Your program should continue reading values until the user enters 0. Then it should display all of the values entered by the user (except for the 0) in order from smallest to largest, with one value appearing on each line. Use either the sort method or the sorted function to sort the list
2. When analysing data collected as part of a science experiment it may be desirable to remove the most extreme values before performing other calculations. Write a function that takes a list of values and an non-negative integer, n, as its parameters. The function should create a new copy of the list with the n largest elements and the n smallest elements removed. Then it should return the new copy of the list as the function’s only result. The order of the elements in the returned list does not have to match the order of the elements in the original list.

Write a main program that demonstrates your function. Your function should read a list of numbers from the user and remove the two largest and two smallest values from it. Display the list with the outliers removed, followed by the original list. Your program should generate an appropriate error message if the user enters less than 4 values.

1. Write a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were entered. For example, if the user enters:

first

second

first

third

second

then your program should display:

first

second

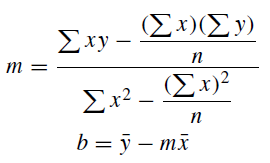
third

1. An integer, n, is said to be perfect when the sum of all of the proper divisors of n is equal to n. For example, 28 is a perfect number because its proper divisors are 1, 2, 4, 7 and 14, and 1 + 2 + 4 + 7 + 14 = 28.

Write a function that determines whether or not a positive integer is perfect. Your function will take one parameter. If that parameter is a perfect number then your function will return true. Otherwise it will return false. In addition, write a main program

that uses your function to identify and display all of the perfect numbers between 1 and 10,000.

1. A line of best fit is a straight line that best approximates a collection of n data points. In this exercise, assume that each point in the collection has an x coordinate and a y coordinate. The symbols  and  are used to represent the average x value in the collection and the average *y* value in the collection respectively. The line of best fit is represented by the equation *y* = *mx* + *b* where *m* and *b* are calculated using the following formulas.



Write a program that reads a collection of points from the user. The user will enter the x part of the first coordinate on its own line, followed by the y part of the first coordinate on its own line. Allow the user to continue entering coordinates, with the x and y parts each entered on their own line, until your program reads a blank line for the x coordinate. Display the formula for the line of best fit in the form y = mx +b by replacing m and b with the values you calculated using the preceding formulas.

For example, if the user inputs the coordinates (1, 1), (2, 2.1) and (3, 2.9) then your program should display y = 0.95x + 0.1.

1. Write the following function that merges two sorted lists into a new sorted list:

**def merge(list1, list2):**

Write a test program that prompts the user to enter two sorted lists and displays the merged list. Here is a sample run:



1. Write the following function that partitions the list using the first element, called a pivot:

**def partition(lst):**

After the partition, the elements in the list are rearranged so that all the elements before the pivot are less than or equal to the pivot and the element after the pivot are greater than the pivot. The function also returns the index where the pivot is located in the new list. For example, suppose the list is [5, 2, 9, 3, 6, 8]. After the partition, the list becomes [3, 2, 5, 9, 6, 8]. Implement the function in a way that takes len(lst) comparisons. Write a test program that prompts the user to enter a list and displays the list after the partition. Here is a sample run:



1. Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time, as shown in the sample run. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue playing. Create a list to store the words, as follows:

# Use any words you wish

**words = ["write", "that", "program", ...]**

****

1. A school has 100 lockers and 100 students. All lockers are closed on the first day of school. As the students enter, the first student, denoted S1, opens every locker. Then the second student, S2, begins with the second locker, denoted L2, and closes every other locker. Student S3 begins with the third locker and changes every third locker (closes it if it was open, and opens it if it was closed). Student S4 begins with locker L4 and changes every fourth locker. Student S5 starts with L5 and changes every fifth locker, and so on, until student S100 changes L100.

After all the students have passed through the building and changed the lockers, which lockers are open? Write a program to find your answer.

(Hint: Use a list of 100 Boolean elements, each of which indicates whether a locker is open (True) or closed (False). Initially, all lockers are closed.)

1. Write a program that identifies all of the words in a string entered by the user. Begin by writing a function that takes a string of text as its only parameter.Your function should return a list of thewords in the string with the punctuation marks at the edges of the words removed. The punctuation marks that you must remove include commas, periods, question marks, hyphens, apostrophes, exclamation points, colons, and semicolons. Do not remove punctuation marks that appear in the middle of a words, such as the apostrophes used to form a contraction. For example, if your function is provided with the string "Examples of contractions include: don’t, isn’t, and wouldn’t." then your function should return the list ["Examples", "of", "contractions", "include", "don’t", "isn’t", "and", "wouldn’t"].

Write a main program that demonstrates your function. It should read a string from the user and display all of the words in the string with the punctuation marks removed.