List:

1.Take two lists, say for example these two:

A **=** [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

B **=** [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

And write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes

1. Ask the user for a string and print out whether this string is a palindrome or not. (a **palindrome** is a string that reads the same forwards and backwards.)
2. Given a string of odd length greater 7, return a string made of the middle three chars of a given string
3. Given 2 strings, s1, and s2 return a new string made of the first, middle and last char each input string
4. Arrange string characters such that lowercase letters should come first
5. Given a string input Count all lower case, upper case, digits, and special symbols
6. String characters balance Test : We’ll say that a String s1 and s2 is balanced if all the chars in the string1 are there in s2. characters position doesn’t matter.
7. Find all occurrences of “USA” in given string ignoring the case:"Welcome to USA. usa awesome, isn't it?"
8. Given a string, return the sum and average of the digits that appear in the string, ignoring all other characters

For Example: –sumAndAverage("English = 78 Science = 83 Math = 68 History = 65") = sum 294 Percentage is 73.5

10.Given an input string, count occurrences of all characters within a string

For Example: count("pynativepynvepynative") = {'p': 3, 'y': 3, 'n': 3, 'a': 2, 't': 2, 'i': 2, 'v': 3, 'e': 3}

11.Given two strings, s1 and s2, create a mixed String

Note: create a third-string made of the first char of the last char of b, the second char of the second last char of b, and so on. Any leftover chars go at the end of the result.