Strings Questions

1. Write a function that tests whether a string is a palindrome.
2. Write a function that takes a list of strings an prints them, one per line, in a rectangular frame. For example the list ["Hello", "World", "in", "a", "frame"] gets printed as:

\*\*\*\*\*\*\*\*\*

\* Hello \*

\* World \*

etc

\*\*\*\*\*\*\*\*\*

1. Write function that translates a text to Pig Latin and back. English is translated to Pig Latin by taking the first letter of every word, moving it to the end of the word and adding ‘ay’. “The quick brown fox” becomes “Hetay uickqay rownbay oxfay”.
2. Write a program that finds the longest palindromic substring of a given string. Try to be as efficient as possible!
3. Given an input string "aabbccba", find the shortest substring from the alphabet "abc".   In the above example, there are these substrings "aabbc", "aabbcc", "ccba" and "cba". However the shortest substring that contains all the characters in the alphabet is "cba", so "cba" must be the output.   Output doesnt need to maintain the ordering as in the alphabet.   Other examples:  input = "abbcac", alphabet="abc" Output : shortest substring = "bca".
4. Given a string, print out all of the unique characters and the number of times it appeared in the string
5. Given two strings a and b, find whether any anagram of string a is a sub-string of string b. For eg:  if a = xyz and b = afdgzyxksldfm then the program should return true.
6. [Given a string, find whether it has any permutation of another string. For example, given "abcdefg" and "ba", it shuold return true, because "abcdefg" has substring "ab", which is a permutation of "ba".](https://www.careercup.com/question?id=15555796)
7. [Code a function that receives a string composed by words separated by spaces and returns a string where words appear in the same order but than the original string, but every word is inverted.  Example, for this input string](https://www.careercup.com/question?id=5106757177180160)
8. Given a string (1-d array) , find if there is any sub-sequence that repeats itself.  Here, sub-sequence can be a non-contiguous pattern, with the same relative order.   Eg:   1. abab <------yes, ab is repeated  2. abba <---- No, a and b follow different order  3. acbdaghfb <-------- yes there is a followed by b at two places  4. abcdacb <----- yes a followed by b twice   The above should be applicable to ANY TWO (or every two) characters in the string and optimum over time.   In the sense, it should be checked for every pair of characters in the string.
9. Given two strings, write a program that efficiently finds the longest common subsequence.
10. Given two strings, write a program that outputs the shortest sequence of character insertions and deletions that turn one string into the other.