**Longest Palindromic Substring**

The  Longest Palindromic Substring (LPS) is the problem of finding a maximum–length contiguous substring of a given string that is also a palindrome (回文). For example, the longest palindromic substring (LPS) of “bananas” is “anana”. Here a string is said to be palindrome if reverse of the string is same as string. For example, “abba” is palindrome, but “abbc” is not palindrome. Note that the longest palindromic substring is not guaranteed to be unique; for example, in the string “abracadabra”, there is no palindromic substring with length greater than three, but there are two palindromic substrings with length three, namely, “aca” and “ada”. Given a string s, your task in this problem is to write a program that can find and print out the longest palindromic substring in s. Here lists some examples. If the input string s is “babad”, the LPS is either “bab” or “aba”. Another example is: if the input string s is “cbbd”, the LPS is “bb”.

In this problem, you will be able to practice the conditionals (if, else), the while loop (and continue, break), logic operations (and, or, not), relational operations (>, =, <=,  ==, !=), string index and slicing, or built‐in string functions, such as len().

