**Palindromic patitioning / Minimum Palindromic Cut**

Given a string **str**, a partitioning of the string is a *palindrome partitioning* if every sub-string of the partition is a palindrome. Determine the fewest cuts needed for palindrome partitioning of given string.

**Example 1:**

**Input:** str = "ababbbabbababa"

**Output:** 3

**Explaination:** After 3 partitioning substrings

are "a", "babbbab", "b", "ababa".

**Example 2:**

**Input:** str = "aaabba"

**Output:** 1

**Explaination:** The substrings after 1

partitioning are "aa" and "abba".

**Your Task:**  
You do not need to read input or print anything, Your task is to complete the function **palindromicPartition()** which takes the string str as input parameter and returns minimum number of partitions required.

**Expected Time Complexity:** O(n\*n) [n is the length of the string str]  
**Expected Auxiliary Space:** O(n\*n)

**Constraints:**  
1 ≤ length of str ≤ 500