Python Recursion Workshop

CSCI 204

Solve the following problems using recursion. You can work in pair or alone. Use whichever Python environment you feel comfortable.

1. Write a function named `remove_duplicates(str)` to remove consecutive duplicates from a string recursively. For example,
   `remove_duplicates(“aabccba”)` would return `“abcba”`
   `remove_duplicates(“abc”)` would return `“abc”`
   `remove_duplicates(“”)` would return `“”`
2. Write a function to determine if a integer is odd or not. For example
   `is_odd(3)` would return `True`
   `is_odd(0)` would return `False`
   `is_odd(12)` would return `False`
   Hint: you can use two base cases, 0 is not an odd number, 1 is an odd number.
3. Determine if a string is a palindrome. A string is a palindrome if it reads the same forward and backward. The input string should be considered as case insensitive, i.e., ‘abc’ is the same as ‘Abc’.
   `is_palindrome(‘aba’)` returns `True`
   `is_palindrome(‘a’)` returns `True`
   `is_palindrome(‘ab’)` returns `False`
   `is_palindrome(‘Able was I ere I saw Elba’)` returns `True`
4. Given the following binary search algorithm,

   ```
   def  bin_search(nums, target, left, right):
       found = False
       mid = (left + right) // 2
       while (not found and left <= right):
           if nums[mid] == target:
               found = True
           elif nums[mid] < target:  # search for upper half
               left = mid + 1
           else:
               right = mid -1        # search for lower half
           mid = (left + right) // 2
       if found:
           return mid
       else:
           return -1
   ```

   Re-write the function in recursive form.