Problems based on Recursion - 7

Assignment Solutions

PW SKILLS

Q1 - Given a string, remove adjacent duplicates characters from it. In other words, remove all consecutive same characters except one using recursion. (Easy)

Input: aabbbbcdd
Output: abcd

Input 2: abbcddaaaerr
Output 2: abcdaer

Code link: https://pastebin.com/6d4Z8r1v

```
aavdssseedf
avdsedf
PS C:\Users\atiba\OneDrive\Desktop\C++programs>
```

Explanation:

- We have created a function ***"removeAdjacentDuplicates"*** that takes 3 parameters as input. One is the original string, one will be our **answer** string which we will be passing by reference since we want all the function calls to make changes in our **answer** string and not in their temporary strings and last would be the index.
- If the index has already reached the length of the string we need to terminate the code that is why we write a return statement.
- The return type of this function is void since we do not want this function to return anything.
- If the base case is not hit then we need to check that if the last element/character of the answer string is equal to the current element at which the ***index*** is pointing then we need not to consider this element as a part of our answer string because keeping this character in ***answer*** will again lead to adjacent duplicates.
- So we simply checked if the last element of the answer string is not equal to ***s[index]*** then we will simply add this to our final string.
- And we will make a recursive call to the remaining indices of the string.
- Hence we made a recursive call with parameters ***s , index + 1 , answer***.

Q2 - Write a recursive program to efficiently print the reverse of a given string in C++. (Easy)

Input 1: "collegewallah"
Output 1: "hallawegelloc"

Input 2: "pwskills"
Output2: "sllikswp"

Code Link: https://pastebin.com/egqtBZVZ

```
hallawegelloc

...Program finished with exit code 0
Press ENTER to exit console.
```

Explanation:

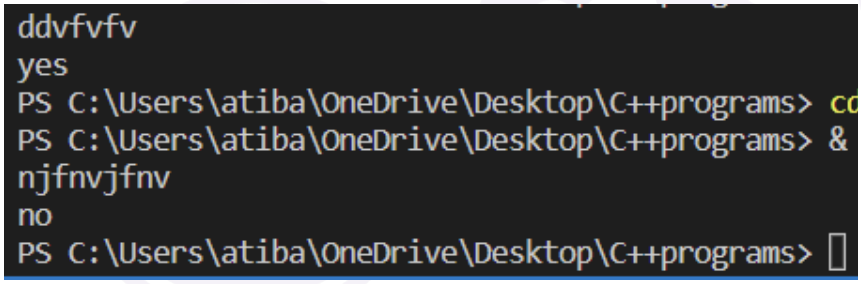
- We have created a function “reverse” where in each function call we have stored a character variable which contains the character at **index** in the function stack .
- In the next recursive call we store the character at **index** in the function stack and keep doing this till the base case is reached.
- Once the base condition is satisfied we return and start printing the character stored in the function stack.
- This will print the reverse of the string.
  For example: We have a string “abcd” the function stack will be
  d -> fourth call
  c ->third call
  b ->second call
  a -> first call

Q3 - Given a string, check if it is a rotated palindrome or not using recursion. “baabcc” is a rotated palindrome as it is a rotation of palindrome “abccba”. (Medium)

Input 1: cbaabcd
Output 1 : yes
Explanation : if we rotate this string 3 times it will turn out to be abcdcba which is a palindrome.

Input 2 : abcdedca
Output 2 : no

Code Link: https://pastebin.com/2c7qNUHs



Explanation:

- From the main function we have called a function “isRotatedPalindrome” which is of boolean type that will return a true if the current string is a palindrome.
- If the current string is not a palindrome then the function “isRotatedPalindrome” will rotate the string by adding the last character of the current string to the front of the string and removing the last character in each iteration.
- The rotated string will again be checked if it is a palindrome or not.
- In this way we will rotate the string n times where n is equal to the size of the string and check for each rotated string if it is a palindrome or not.

Q4 - Write a program to merge 2 strings alternately using recursion. (Medium)

Input1:
abcd
efgh
Output1:
aebfcgdh

Code Link: https://pastebin.com/NcahPre3

```
aebfcgdh

...Program finished with exit code 0
Press ENTER to exit console.
```

Explanation:

- We have a function named cat which takes in 5 arguments **string a, string b, index1 and index2** which represents the index at which they point to strings a and b respectively and a string s which is the merged string.
- We will add the index1 char of string a and index2 char of string b to string s and then increase both index1 and index2 until the base condition is reached.
- If both index1 and index2 reach the end of their strings then we simply return.
- If index1 or index2 reach at the end of their strings then we just add the remaining string to the string s.