



Recursion-VII

Important concepts of DSA



Recursion-VII

- There are few more questions on recursion which clears the idea how the recursion works
- Like, replace character recursively, remove duplicate characters recursively

Remove duplicates recursively

- In a string, when there are duplicates which are consecutive, they can be removed
- Generally, after last character, there is '**\0**' which marks the end of the string or a character array
- If one tries to access the element when the string ends, then it will be '**\0**'
- And also, the function should be of void type

Code:

```
void removeConsecutiveDuplicates(char *input)
{
    if(input[0]=='\0')
        return ;
    int n = strlen(input);
    if(input[0]==input[1])
    {
        for(int i=0;i<n-1;i++)
        {
            input[i] = input[i+1];
        }
        input[n-1]='\0';
        removeConsecutiveDuplicates(input);
    }
    else
        removeConsecutiveDuplicates(input+1);
}
```

Input:

aabccba

Output:

abcba

Input:

xxxyyyzwwzzz

Output:

xyzwz

- The base condition here is to check whether the string is over or not. If the char array is over, the "**return ;**" will throw the flow out of the function
- Now, the condition is to check whether the characters which are coming consecutively are equal or not
- If they are same, then one will have to shift the character to the next index and one will have to set the last character to '**\0**'
- If the characters are "**xyy**", after shift, it would be like "**xyy**" before adding the '**\0**'

- After adding null character, it would be "**xy**" and also, it's length will be reduced to 2
- And now, one'll have to call function with input only but this time input is having the new "**input**" as there were many changes made in the "**input**" and it's length was also reduced by one
- Now, what if the characters are not same, then one'll have to check for the string that is upcoming and hence the function called will be the same but this time, parameter will be "**input+1**" which is showing the part that is after the 0th index of input
- It will be equivalent to "**yz**" if the value of input is "**xyz**"

**If you have any doubt,
let me know in the comments**

Happy Coding

