

Problems based on Recursion - 7

Assignment Solutions



Q1. Write a program to merge 2 strings alternately using recursion starting from the first input string.

Input:

abcd
efgh

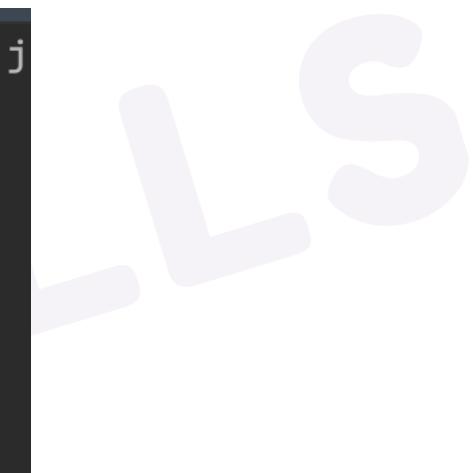
Expected Output:

aebfcgdh

Explanation:

- Make a function merge and pass both the strings as arguments.
- Use blank string ans to store the merged string.
- Base case conditions are if length of string a == 0, then add b to ans and return ans, similarly for b.
- If base conditions are not satisfied, add first character of a to ans followed by first character of b and call the function recursively by passing the arguments as a.substring(1, a.length()), b.substring(1, b.length()), add it to ans also.
- Return ans in the end.

Code: <https://pastebin.com/UPcPTcU0>



```
/Library/Java/JavaVirtualMachines/jdk-19.j
↑ Enter the first string:
↓ abcd
→ Enter the second string:
↓ efgh
← aebfcgdh
Process finished with exit code 0
```

Q2. Given a string, find its first uppercase letter and return the remaining substring, starting from the uppercase letter.

Input:

collegeWallah

Expected Output:

Wallah

Explanation:

- Create a recursive function of String return type with input string and an index variable initialized from 0 as parameters.
- Our base case is if index goes out of bounds and is equal to length of the string, it means we have found no uppercase letter, and just return a blank string "" in this case.
- To check if character at the current index is uppercase, we use an inbuilt method: Character.isUppercase().
- If current character is uppercase, return substring of s from index till end.
- Else call the recursive function with the string and 1 incremented index.

Code: <https://pastebin.com/EUSmbJYT>

```
/Library/Java/JavaVirtualMachines/]
Enter the input string:
collegeWallah
Wallah

Process finished with exit code 0
|
```

Q3. Given a string, count total number of consonants in it. A consonant is an English alphabet character that is not vowel (a, e, i, o and u). Examples of constants are b, c, d, f, and g.

Input:

pwskills

Expected Output:

7

Explanation:

- Create an int return type recursive function with input string and its length as function arguments.
- We recursively traverse the string in a reverse fashion.
- Our base case is to check if $n==1$, this means we are at the first character of the string, now if this character is a consonant, return 1 else return 0.
- To check if a character is a consonant, we create another function just so we don't need to write the checks again and again.
- Create another function is consonant, and pass a character as the argument.
- First convert the character to lowercase as the input string can have lowercase and uppercase letters both.
- Now check if character is not any of the vowels, if not, return true, else false.
- After the base case in the recursive function, check if character at $n-1$ index is consonant, if so, return the recursive function with string and $n-1$ as arguments and add 1 to this.
- If it is not a consonant, return the recursive function with string and $n-1$ as arguments.

Code: <https://pastebin.com/kwtqXXtN>

```
/Library/Java/JavaVirtualMachines/]
Enter the input string:
pwskills
7

Process finished with exit code 0
|
```

Q4. Given a string, return the number of lowercase characters in it using recursion.

Input:

CollegeWallah

Expected Output:

11

Explanation:

- Create an int return type recursive function with input string and current index initialized with 0 as function arguments.
- Our base case is to check if index is of the last character of the string, this means we are at the last character of the string, now if this character is a lowercase, return 1 else return 0.
- To check if a character is a lowercase, we can check either by using the inbuilt method, Character.isLowercase() or as in here we check it by ASCII codes.
- ASCII codes are the numeric values unique to each character.
- Lowercase letters from a to z lie in the range of 97('a') to 122('z').
- If the value of character lies in this range, it means that the character is lowercase.
- After the base case in the recursive function, check if character at index is lowercase, if so, return the recursive function with string and index+1 as arguments and add 1 to this.
- If it is not lowercase, return the recursive function with string and index+1 as arguments.

Code: <https://pastebin.com/GjK5kMCG>

```
/Library/Java/JavaVirtualMachines/]
Enter the input string:
CollegeWallah
11

Process finished with exit code 0
```

Q5. Given a string, recursively implement atoi() or Integer.parseInt() method on it without actually using the method.

Input:

298

Expected Output:

298

Explanation:

- atoi() function of C++ or Integer.parseInt() method of java takes a string (which represents an integer) as an argument and returns its value so it looks the same but is of different data type.
- To convert a string to integer, we need to pick the first character, convert it into integer by subtracting '0' character from it as '0' has the initial ASCII code value and serial numbers have consecutive ASCII values as in the decimal system, all values are set in relation to 0.
- For example, atoi(982) is calculated in following manner:

```
"982"  
 / \  
9 "82"  
 / /\  
90 8 "2"  
 \ / /\  
98 2 ""  
 |  
982
```

- We keep picking the characters from start of the string, multiply current ans by 10 and then add the next character to ans.
- To do it recursively, we do it in a reverse fashion.
- We call a recursive function for string from first index to second last index, and add last digit to this ans.
- We create a recursive function with string and index as length of string as its arguments.
- Our base case is if n==0, which means we are the first character, so just convert it to digit and return.
- Call the recursive function with string and n-1 as arguments, multiply this with 10 and add last digit to this and return.

Code: <https://pastebin.com/wZ6ficT2>

```
/Library/Java/JavaVirtualMachines/  
Enter the input string:  
298  
298  
  
Process finished with exit code 0
```