Lesson:

Problems based on Recursion - 7

Pre-Requisites

- Recursion basics
- Working rules of recursive functions

List of Concepts Involved

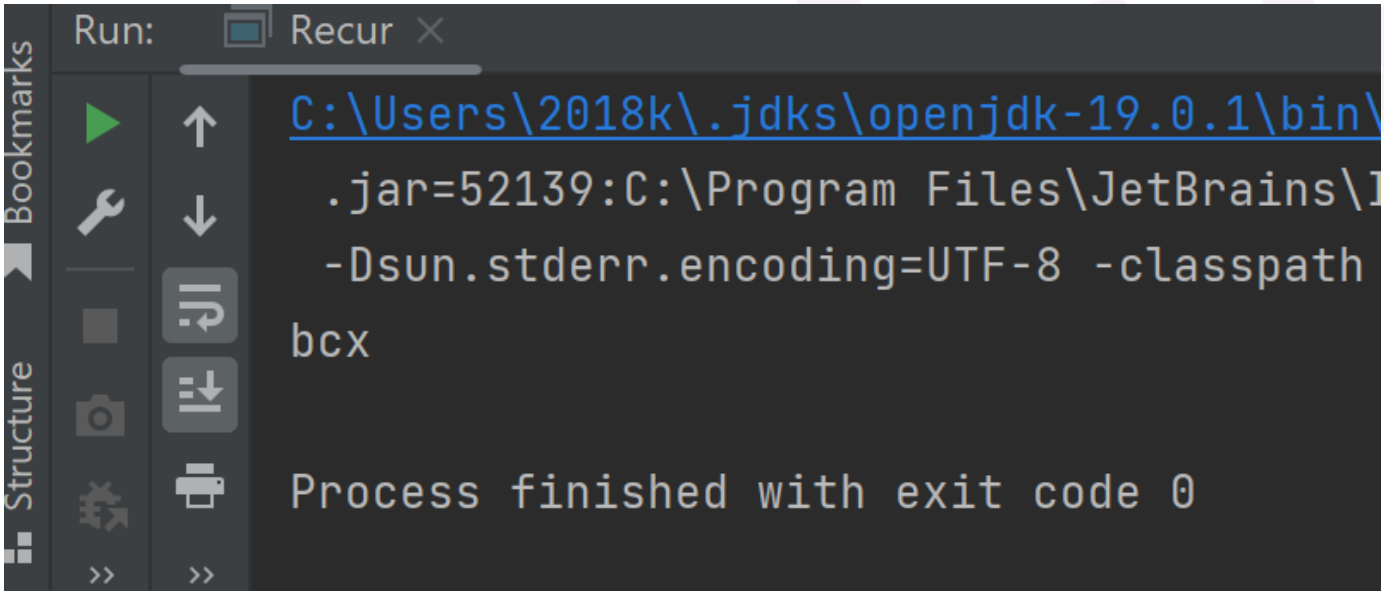
- Recursion on strings

Topic : Recursion on strings

The method doesn't differ much from the recursion used in arrays.
Let's see some questions

Problem 1 : Remove all the occurrences of 'a' from string s = "abcax".

Code: https://pastebin.com/pAaAZBsD



Explanation:

- To solve this problem recursively, I have created one method removeChar(). It takes two arguments, one is a string and the second is the character which is to be removed.
- The method *removeChar()* calls itself recursively until the base or terminating condition is not reached.
- The base condition is when the length of the string is equal to zero. Until the string is not empty it calls itself.
- If the first character of the string matches with ch(character which is to be removed),then call method removeChar(), and pass the substring starting from index 1.
- If the first character of string does not match with ch then ,Add the first character of str and call method removeChar(), and pass the substring starting from index 1.

Problem 2 : Write a program to print the reverse of a string using recursion.
Input str="Physics"
Output = scisyhP

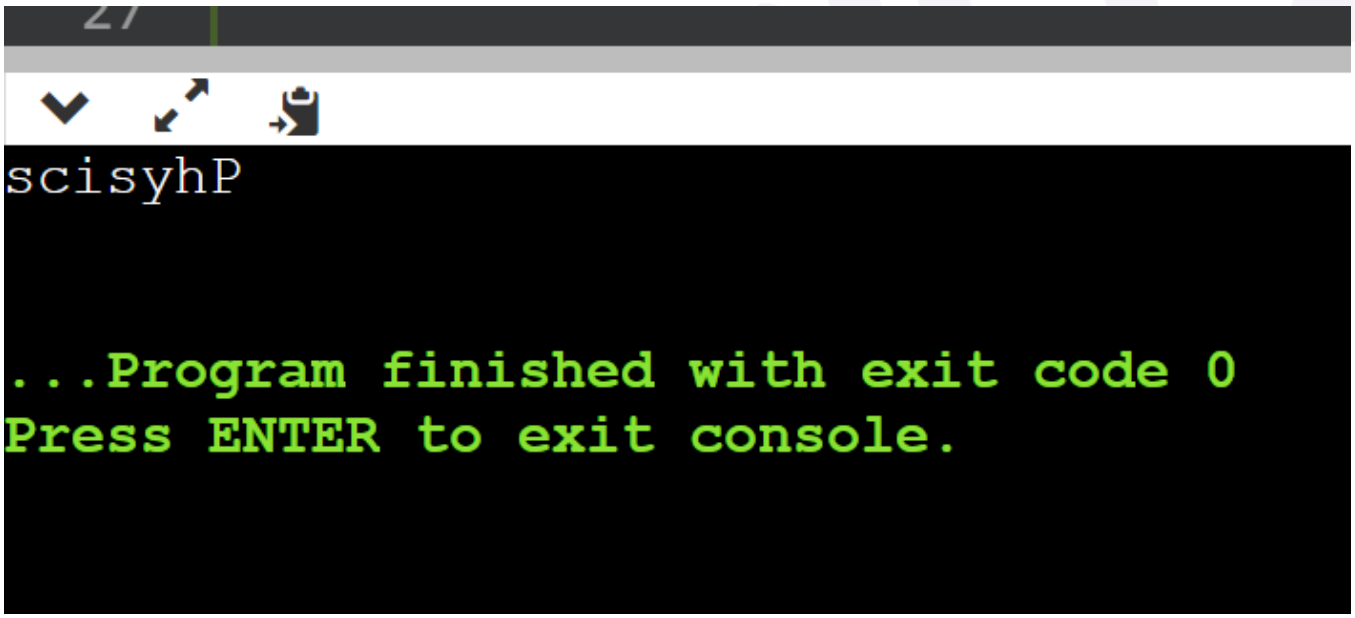
- Here we have created a method named reverse() which will return the reversed string. So its return type is String.
- In this method, first, we have checked that the string is empty or not. If the string is empty or the size of the string is 1 then it returns the same string because the reverse of the string having size 1 is the same as the original string . This is the base case condition for this function.
- Here, we have used the following two methods of the String class:

substring(): It returns the new string that is a substring of the specified string. It parses a parameter that specifies the starting index (beginning) of the substring.

charAt(): It returns a character at the specified index. The index lies between 0 to length()-1.

- If the base case condition is false then this method parses a substring returned by the substring() method.
- We recursively call the reverse function and pass the substring starting from index 1, then we add the first character at the end of the string returned by function reverse(str.substring(1)).

Code: https://pastebin.com/kJaTLPJW



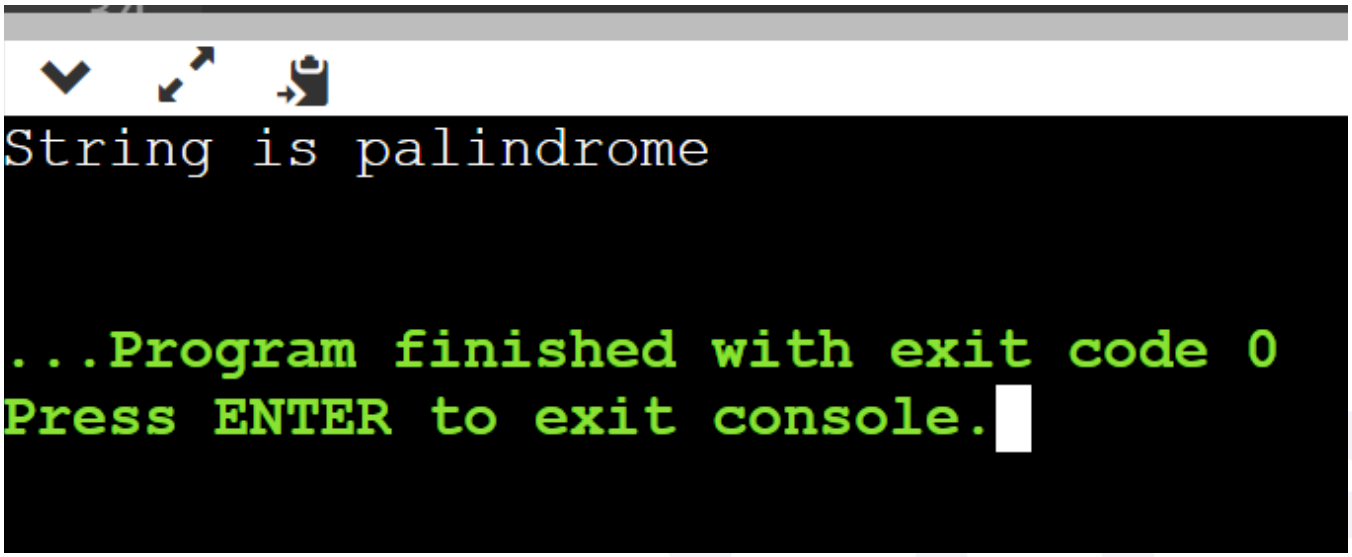
Problem 3 : Write a program to check whether a string is a Palindrome.
Input : `str = "abcba"`
Output: `Yes`

Explanation

- A string is said to be a palindrome if it is the same if we start reading it from left to right or right to left. So in the above example str is the same if we read it either from left to right or right to left.
- Now the task is just to find out if its reverse string is the same as it is.

- We have seen in a previous question how to find the reverse of a string, let's find the reverse and check if the reversed string is the same as the original one then we can say the string is palindrome otherwise no.

Code: https://pastebin.com/02R2MSzU



Upcoming Class Teasers:

- Problems based on recursion.