ASSIGNMENT 3

String Operations

Due Date: Friday, 14th Oct., 11:59 PM PDT

You will create a class that performs different functions on Strings that are sent to it.
All the methods you create will be *static*. Only two methods (mentioned below) and the main method need to be *public*. Any other methods that you write to help these methods should be *private* because they are only used internally within the class.

Requirements:

1. Prompt users to enter 3 string inputs. Consider strings with alphabetic characters (a - z, A - Z) and/or numbers (0 - 9) as valid*. Print the following usage when an input string is invalid or is empty.

```
Usage: Enter a string that contains alphabetic characters and numbers
```

2. Create a method "**isPalindrome**" that takes a String as a parameter and returns true if the given String parameter is a palindrome and false if it is not. In your main method, this method should be call with **the first user input** and print either "*The given string is a palindrome."* or *"The given string is not a palindrome.*" as output based on the returned value.

> A word is a palindrome if it reads the same forwards and backwards. For example, the word "level" is a palindrome. The idea of a palindrome can be extended to phrases or sentences if we ignore details like punctuation. Here are two familiar examples:
>
> *Madam, I'm Adam*
>
> *A man, a plan, a canal: Panama*
>
> We can recognize these more elaborate examples as palindromes by considering the text that is obtained by removing all spaces and punctuation marks and converting all letters to their lower-case form.

* Strings contain only non-word characters are invalid.

Madam, I'm Adam ==> madamimadam

A man, a plan, a canal: Panama ==> amanaplanacanalpanama

If the "word" obtained from a phrase in this manner is a palindrome, then the phrase is a palindrome. Your method should also ignore the case of the letters. A palindrome is determined by considering only alphabetic characters (a - z, A - Z) and numbers (0 - 9) as valid text.

3. Create another method named "**findSubstring**" that takes 3 strings as parameters and checks if the first string contains a substring that starts with the second string, ends with the third string, **and** has an even number of characters in between[†]. Return the substring if it exists and return null if not. This method will be called in your main method with all three user inputs and print either "*The substring is <substring>"* or "*No such substring in the given string.*"[‡] as output based on the returned value. Your method should ignore the letter case and assume that no more than one such substring exists in the initial string.

4. Your output should be like the following test samples.

Sample Test Cases:

Test Case 1:

```
Please enter a string:
ATGCGATAC6TA
Please enter a string:
atg
Please enter a string:
6ta

The given string is "ATGCGATAC6TA"
The prefix is "atg"
The suffix is "6ta"

The given string is not a palindrome.
```

[†] Count only alphabetic characters and numbers.

[‡] Replace <substring> with corresponding values.

```
The substring is "ATGCGATAC6TA"

Program Completed
```

Test Case 2:

```
Please enter a string:
*^_^*
Usage: Enter a string that contains alphabetic characters and numbers
Please enter a string:
Was it a car or a cat i saw?
Please enter a string:
car
Please enter a string:
cow

The given string is "Was it a car or a cat i saw?"
The prefix is "car"
The suffix is "cow"

The given string is a palindrome.
No such substring in the given string.

Program Completed
```

Test Case 3:

```
Please enter a string:
Lewd did I live, & evil I did dwel.
Please enter a string:
did
Please enter a string:
did

The given string is "Lewd did I live, & evil I did dwel."
The prefix is "did"
The suffix is "did"

The given string is a palindrome.
```

```
The substring is "did I live, & evil I did"

Program Completed
```

Notes:

1. The program should always print '**Program Completed**' before exiting.
2. Check if your program works properly by giving different inputs.
3. Give comments to increase code readability.
4. Mention the sources used to complete the assignment.
5. Save the file as **LastnameFirstnameA3**.java. (DO NOT submit files with names such as "LastnameFirstnameA3-1")
6. Submit **.java file** only.