Make sure your prompts and printouts match EXACTLY to the specification (INCLUDING capitalization, punctuation, spaces and newlines)!

Similarly, your java files / class names / method names MUST match the spelling and capitalization EXACTLY!

You can use the checkboxes to track whether you've met each requirement.

#	***Requirements***	
1	**Create a class named FancyTitle with the following public static methods:**	
	public static int countFibSteps(int maxVal)	
	This method will compute and return the number of Fibonacci sequence steps necessary to get to *maxVal*.	

The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, and so on. The next number in the sequence is computed as the last number plus the second-to-last number. For example,
$1 + 1 = 2$,
$1 + 2 = 3$,
$2 + 3 = 5$, etc.

For this method, we'll start with the current and previous numbers equal to 1. Thus, if *maxVal* is 1 or less, the number of steps is 0.

Start steps at 0.

While the current number is less than *maxVal*:

- Set $newVal = current + previous$
- Set $previous = current$
- Set $current = newVal$
- Increment the steps

Return steps at the end of the method.

Examples:

- If $maxVal == -1$, $steps = 0$
- If $maxVal == 1$, $steps = 0$
- If $maxVal == 2$, $steps = 1$
- If $maxVal == 5$, $steps = 3$
- If $maxVal == 20$, $steps = 6$

---

public static String repeatChar(char c, int cnt)

Given a character *c*, create and return a String with that character repeated *cnt* number of times.

Example: if $c ==$'*' and $cnt == 5$, a String "*****" will be returned.

---

public static String [] makeFibStringList(char c, int levels)

The purpose of this method is to create and return an array of Strings with *levels* number of elements, where each element in the array is a String that has the character *c* repeated by the next Fibonacci number.

For example, if $c ==$'*' and $levels == 5$, then the resulting String array should contain 5 Strings:

```
"*",
"**",
"***",
"*****",
"********"
```

Allocate space for an array of Strings with *levels* number of elements.

Set current and previous numbers at 1.

For each level:

- Use your repeatChar() method to generate the correct String with character *c* and *current* number of characters.
- Store this String in the array.
- Compute the next Fibonacci number as you did in countFibSteps().

Return the array of Strings.

public static String makeFilledCenterString(String border, int desiredLen)

The purpose of this method is to create and return a String with *border* at the start and end, and with spaces in between such that the total length is *desiredLen*.

For example, if $border ==$"****" and $desiredLen == 13$, then the returned String will be "**** ****" (the total length of the String is 13; five spaces are in between the border Strings).

Get the length of *border*.

Compute the number of spaces necessary as $numSpaces = desiredLen - 2 * (length\ of\ border)$.

Use your repeatChar() method to create the String of spaces.

The full String will be $border + spaces + border$.

Return this full String.

public static String [] makeFooter(char c, int desiredLen)

The purpose of this method is to create and return an array of Strings for the title card's footer.

For example, if c ==`"*"` and $desiredLen == 20$, then the array will have 5 Strings:

```
"*                   *",
"**                 **",
"***               ***",
"*****           *****",
"********     ********"
```

Each half (that doesn't have spaces) is generated using makeFibStringList(), so you must first calculate the number of levels as countFibSteps(desiredLen/2).

Once you have the number of levels, use makeFibStringList() to get an array of String for the border part.

Loop through each String in your array and use makeFilledCenterString() to appropriately change each String.

Return your updated array of Strings.

---

public static String createFancyTitle(String message, char c)

The purpose of this method is to take a message (with a specified border character) and create a SINGLE String with a fancy title card.

NOTE: This method does NOT print anything out! It only RETURNS the String!

For example, if the message is "A few moments later..." and the border character is "*", then the resulting title card String should be:

```
"*************************\n" +
"********          ********\n" +
"*****                *****\n" +
"***                    ***\n" +
"**                      **\n" +
"*                        *\n" +
"* A few moments later... *\n" +
"*                        *\n" +
"**                      **\n" +
"***                    ***\n" +
"*****                *****\n" +
"********          ********\n" +
"*************************\n"
```

Compute the desired length as the length of the message plus 4 (to account for the extra 2 border characters and 2 spaces around the message).

Create the footer String array using makeFooter().

Start with an empty String *output*.

Append a line of the border character of the computed desired length.

Append the HEADER lines by looping through the footer array BACKWARDS.

Append the border character + a space + the message + a space + the border character + a newline.

Append the lines of the footer.

Again, append a line of the border character of the computed desired length.

	Append the HEADER lines by looping through the footer array BACKWARDS.
	Append the border character + a space + the message + a space + the border character + a newline.
	Append the lines of the footer.
	Again, append a line of the border character of the computed desired length.
	Be sure to append appropriate newlines (there also should be a newline after the very last line).
	Return your *output* String.

Create a class named TitleProgram with a main() method that does the following:

	Create a Scanner object to read from System.in.	
	Print out "Enter message:" using System.out.println().	
	Read in a String LINE using the nextLine() method of your Scanner object and put it into a String variable *message*.	
	Call FancyTitle.createFancyTitle(message, "*") and print the returned title card with System.out.println().	
	Call FancyTitle.createFancyTitle(message, "$") and print the returned title card with System.out.println() (note the different border character).	

Sample Runs

TitleProgram: Run 1

```
Enter message:
A few moments later...
**************************
********          ********
*****                *****
***                    ***
**                      **
*                        *
* A few moments later... *
*                        *
**                      **
***                    ***
*****                *****
********          ********
**************************
```

TitleProgram: Run 2

```
Enter message:
So much later that the old narrator got tired of waiting and they had to hire a new one...
*****************************************************************************************
**********************************                      **********************************
********************                                          ********************
*************                                                        *************
********                                                                   ********
*****                                                                         *****
***                                                                           ***
**                                                                             **
*                                                                               *
* So much later that the old narrator got tired of waiting and they had to hire a new one... *
*                                                                               *
**                                                                             **
***                                                                           ***
*****                                                                         *****
********                                                                   ********
*************                                                        *************
********************                                          ********************
**********************************                      **********************************
*****************************************************************************************
```