Homework1

This question involves analyzing and modifying a string. The following *Phrase* class maintains a phrase in an instance variable and has methods that access and make changes to the phrase. You will write the complete *Phrase* class. Note that when writing these methods, use the methods you have already written rather than writing out code from scratch to perform the same function. Be sure to test all of your methods with a psvm.

Partial *Phrase* class file

```
class Phrase {

    private String currentPhrase;

        public Phrase(String p)
        {
                currentPhrase = p;
        }
        public int findNthOccurrence(String str, int n)
        {
                // implement this method
        }
        public void replaceNthOccurrence(String str, int n, String repl)
        {
                // implement this method
        }
        public int findLastOccurrence(String str)
        {
                //  implement this method
        }
        public String toString()
        {
                return currentPhrase;
        }
}
```

Write the *Phrase* method *findNthOccurrence(String str, int n)*, which will return the index of the nth occurrence of the string *str*. If the string *str* does not exist, the method returns *-1*. In addition, if the *nth* string *str* does not exist, the method returns -1.

Consider a few examples of using findNthOccurrence.

Phrase joe = new Phrase("Mary had a little lamb");

method call	value returned
System.out.println(joe.findNthOccurrence("a", 5);	-1
System.out.println(joe.findNthOccurrence("a", 4);	19
System.out.println(joe.findNthOccurrence("z", 1);	-1
System.out.println(joe.findNthOccurrence("had", 1);	5

Write the *Phrase* method *replaceNthOccurrence*, which will replace the n*th* occurrence of the string *str* with the string *repl*. If the n*th* occurrence does not exist, *currentPhrase* remains unchanged.

Several examples of the behavior of the method *replaceNthOccurrence* are shown below.

Code segments	Output produced

`Phrase phrase1 = new Phrase("A cat ate late.");`
`phrase1.replaceNthOccurrence("at", 1, "rane");`
`System.out.println(phrase1);`
	`A crane ate late.`

`Phrase phrase2 = new Phrase("A cat ate late.");`
`phrase2.replaceNthOccurrence("at", 6, "xx");`
`System.out.println(phrase2);`
	`A cat ate late.`

`Phrase phrase3 = new Phrase("A cat ate late.");`
`phrase3.replaceNthOccurrence("bat", 2, "xx");`
`System.out.println(phrase3);`
	`A cat ate late.`

`Phrase phrase4 = new Phrase("aaaa");`
`phrase4.replaceNthOccurrence("aa", 1, "xx");`
`System.out.println(phrase4);`
	`xxaa`

`Phrase phrase5 = new Phrase("aaaa");`
`phrase5.replaceNthOccurrence("aa", 2, "bbb");`
`System.out.println(phrase5);`
	`abbba`

Write the Phrase method *findLastOccurrence*. This method finds and returns the index of the last occurrence of a given string in *currentPhrase*. If the given string is not found, -1 is returned. The following tables show several examples of the behavior of the method *findLastOccurrence*.

Phrase phrase1 = new Phrase("A cat ate late.");

Method call	Value returned
phrase1.findLastOccurrence("at")	11
phrase1.findLastOccurrence("cat")	2
phrase1.findLastOccurrence("bat")	-1