

AP® COMPUTER SCIENCE A 2017 SCORING GUIDELINES

Question 3: PhraseEditor

Part (a)	<code>replaceNthOccurrence</code>	5 points
-----------------	-----------------------------------	-----------------

Intent: Replace the *n*th occurrence of a given string with a given replacement

- +1** Calls `findNthOccurrence` to find the index of the *n*th occurrence
- +1** Preserves `currentPhrase` only if *n*th occurrence does not exist
- +1** Identifies components of `currentPhrase` to retain (uses `substring` to extract before/after)
- +1** Creates replacement string using identified components and `repl`
- +1** Assigns replacement string to instance variable (`currentPhrase`)

Part (b)	<code>findLastOccurrence</code>	4 points
-----------------	---------------------------------	-----------------

Intent: Return the index of the last occurrence of a given string

- +1** Calls `findNthOccurrence` to find the index of the *n*th occurrence
- +1** Increments (or decrements) the value used as *n* when finding *n*th occurrence
- +1** Returns the index of the last occurrence, if it exists
- +1** Returns -1 only when no occurrences exist

Question-Specific Penalties

- 1** (q) Uses `currentPhrase.findNthOccurrence`
- 2** (r) Confused identifier instead of `currentPhrase`

AP® COMPUTER SCIENCE A 2017 SCORING GUIDELINES

Question 3: Scoring Notes

Part (a) <code>replaceNthOccurrence</code>			5 points
Points	Rubric Criteria	Responses earn the point if they ...	Responses will not earn the point if they ...
+1	Calls <code>findNthOccurrence</code> to find the index of the <i>n</i> th occurrence	do not use the result of calling <code>findNthOccurrence</code>	
+1	Preserves <code>currentPhrase</code> only if <i>n</i> th occurrence does not exist		fail to use a conditional
+1	Identifies components of <code>currentPhrase</code> to retain (uses <code>substring</code> to extract before/after)	identify start and end of substring to be replaced	
+1	Creates replacement string using identified components and <code>repl</code>		create a replacement string that is out of order
+1	Assigns replacement string to instance variable (<code>currentPhrase</code>)		
Part (b) <code>findLastOccurrence</code>			4 points
Points	Rubric Criteria	Responses earn the point if they ...	Responses will not earn the point if they ...
+1	Calls <code>findNthOccurrence</code> to find the index of the <i>n</i> th occurrence	do not use the result of calling <code>findNthOccurrence</code>	<code>return currentPhrase.lastIndexOf(str);</code> - call <code>findNthOccurrence</code> with an integer parameter of 0
+1	Increments (or decrements) the value used as <i>n</i> when finding <i>n</i> th occurrence	<code>return currentPhrase.lastIndexOf(str);</code> - advance through <code>currentPhrase</code> searching for <i>n</i>th occurrence of <code>str</code>	
+1	Returns the index of the last occurrence, if it exists	<code>return currentPhrase.lastIndexOf(str);</code> - compute the correct value to be returned in all cases, but no return statement exists for any case	- shorten string being searched - always return in first iteration of the loop
+1	Returns -1 only when no occurrences exist	<code>return currentPhrase.lastIndexOf(str);</code>	- compute the correct value to be returned in all cases, but no return statement exists for any case - always return in first iteration of the loop

**AP[®] COMPUTER SCIENCE A
2017 SCORING GUIDELINES****Question 3: PhraseEditor***Part (a)*

```
public void replaceNthOccurrence(String str, int n, String repl)
{
    int loc = findNthOccurrence(str, n);

    if (loc != -1)
    {
        currentPhrase = currentPhrase.substring(0, loc) + repl +
                        currentPhrase.substring(loc + str.length());
    }
}
```

Part (b)

```
public int findLastOccurrence(String str)
{
    int n = 1;
    while (findNthOccurrence(str, n+1) != -1)
    {
        n++;
    }
    return findNthOccurrence(str, n);
}
```