AP® COMPUTER SCIENCE A
2017 SCORING GUIDELINES

Question 3: Scoring Notes

Part (a) `replaceNthOccurrence`			5 points
Points	Rubric Criteria	Responses earn the point if they …	Responses will not earn the point if they …
+1	Calls `findNthOccurrence` to find the index of the nth occurrence	• do not use the result of calling `findNthOccurrence`	
+1	Preserves `currentPhrase` only if nth occurrence does not exist		• fail to use a conditional
+1	Identifies components of `currentPhrase` to retain (uses `substring` to extract before/after)	• identify start and end of substring to be replaced	
+1	Creates replacement string using identified components and `repl`		• create a replacement string that is out of order
+1	Assigns replacement string to instance variable (`currentPhrase`)		
Part (b) `findLastOccurrence`			4 points
Points	Rubric Criteria	Responses earn the point if they …	Responses will not earn the point if they …
+1	Calls `findNthOccurrence` to find the index of the *n*th occurrence	• do not use the result of calling `findNthOccurrence`	• `return currentPhrase.lastIndexOf(str);` • call `findNthOccurrence` with an integer parameter of 0
+1	Increments (or decrements) the value used as *n* when finding *n*th occurrence	• `return currentPhrase.lastIndexOf(str);` • advance through `currentPhrase` searching for *n*th occurrence of `str`	
+1	Returns the index of the last occurrence, if it exists	• `return currentPhrase.lastIndexOf(str);` • 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	• `return currentPhrase.lastIndexOf(str);`	• 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

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);
}
```