AP® COMPUTER SCIENCE A
2018 SCORING GUIDELINES

Question 2: Word Pair

Part (a)	WordPairList	5 points

Intent: *Form pairs of strings from an array and add to an* `ArrayList`

+1 Creates new `ArrayList` and assigns to `allPairs`

+1 Accesses all elements of `words` (*no bounds errors*)

+1 Constructs new `WordPair` using distinct elements of `words`

+1 Adds all necessary pairs of elements from word array to `allPairs`

+1 **On exit:** `allPairs` contains all necessary pairs and no unnecessary pairs

Part (b)	numMatches	4 points

Intent: *Count the number of pairs in an* `ArrayList` *that have the same value*

+1 Accesses all elements in `allPairs` (*no bounds errors*)

+1 Calls `getFirst` or `getSecond` on an element from list of pairs

+1 Compares first and second components of a pair in the list

+1 Counts number of matches of pair-like values

Question-Specific Penalties

-1 (z) Constructor returns a value

Question 2: Scoring Notes

Part (a) `WordPairList`		5 points
Points / **Rubric Criteria**	Responses earn the point if they…	Responses will not earn the point if they…
+1 — Creates new `ArrayList` and assigns to `allPairs`	• `allPairs = new ArrayList();` • `allPairs = new ArrayList<>();` • `this.allPairs = ...`	• initialize a local variable that is never assigned to `allPairs`
+1 — Accesses all elements of `words` (*no bounds errors*)		
+1 — Constructs new `WordPair` using distinct elements of `words`		
+1 — Adds all necessary pairs of elements from word array to `allPairs`	• have a loop bounds error • add unnecessary pairs	• improperly add to an `ArrayList`, e.g., `allPairs.get(i) = x;` • only add consecutive pairs `(words[i], words[i+1])`
+1 — **On exit:** `allPairs` contains all necessary pairs and no unnecessary pairs	• improperly add to an `ArrayList`, e.g., `allPairs.get(i) = x;` • have a loop bounds error	• add pairs (i, i) or (i, j) where i > j
Part (b) `numMatches`		**4 points**
Points / **Rubric Criteria**	Responses earn the point if they…	Responses will not earn the point if they…
+1 — Accesses all elements in `allPairs` (*no bounds errors*)		• access elements of `allPairs` as array elements (e.g., `allPairs[i]`)
+1 — Calls `getFirst` or `getSecond` on an element from list of pairs		
+1 — Compares first and second components of a pair in the list		• compare using ==
+1 — Counts number of matches of pair-like values		• fail to initialize the counter

Return is not assessed in part (b).