**Using and Improving GladLibs**

1.Using the file **dnaMystery1**, which reading frame results in the most unique codons?

0

1

2

2. Using the file **dnaMystery1** with reading frame 2, enter the codon appears exactly 4 times:



3. Using the file **dnaMystery1** with reading frame 1, two codons occur most frequently—that is, 6 times.

Enter one of these two codons.



4. Consider the five files with plays by Shakespeare: **caesar.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt**, and **romeo.txt**.

How many words are there that each occur in all five files?



5. Consider the five files: **caesar.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt**, and **romeo.txt**.

How many words are there that each appear in four of the five files?



6. Consider the five files: **caesar.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt**, and **romeo.txt**.

In which file does the word “sad” NOT appear?

caesar.txt

hamlet.txt

likeit.txt

macbeth.txt

romeo.txt

7. Consider the five files: **caesar.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt**, and **romeo.txt**.

In which files does the word “red” appear? (Choose all that apply.)

caesar.txt

hamlet.txt

likeit.txt

macbeth.txt

romeo.txt

8. In the first version of GladLibs we saw in this lesson, assume an ArrayList named **wordsUsed** will store words that are used as replacements, so no replacement word will be used more than once. Which one of the following could be the MISSING CODE for the code below in GladLibs to make sure that no word is used twice?

String sub = getSubstitute(w.substring(first+1,last));

while (true) {

[MISSING CODE]

}

sub = getSubstitute(w.substring(first+1,last));

if (! wordsUsed.contains(sub)) {

wordsUsed.add(sub);

break;

}

if (! wordsUsed.contains(sub)) {

wordsUsed.add(sub);

break;

}

sub = getSubstitute(w.substring(first+1,last));

if (! wordsUsed.contains(sub)) {

wordsUsed.add(sub);

sub = getSubstitute(w.substring(first+1,last));

break;

}

if ( !wordsUsed.contains(sub)) {

sub = getSubstitute(w.substring(first+1,last));

wordsUsed.add(sub);

break;

}

9. Consider the map version of GladLibs, and consider the method **totalWordsInMap** that returns the total number of words in all the ArrayLists in the HashMap **myMap**. Which one of the following is the best choice for the loop in this method?

for (String category : myMap.keySet()) {

for (int category : myMap.keySet()) {

for (ArrayList<String> category : myMap.keySet()) {

for (ArrayList category : myMap.keySet()) {

10. Consider the map version of GladLibs, and consider the method **totalWordsConsidered** that returns the total number of words in the ArrayLists of the categories that were used for a particular GladLib. What type of private instance variable would be the best choice to solve this problem?

An ArrayList of Strings representing the unique categories used.

An int representing the total number of words from categories.

A HashMap of String to String, mapping a category seen to the first word seen in that category.

An ArrayList of Integers representing the sizes of the unique categories ArrayLists.