Breaking the Caesar Cipher

11/11 points (100%)

Practice Quiz, 11 questions

✔ Congratulations! You passed!

Next Item

1 / 1 points

1.

Which two of the following check correctly if the last character of the String **word** is not a letter?

```
1  if ( ! word.charAt(Character.isLetter(word.length()-1))) {
2      wordlength--;
3  }
```

Un-selected is correct

```
1  if ( ! word.charAt(word.length()-1)) {
2      wordlength--;
3  }
```

Un-selected is correct

```
1  if (word.char(Character.isLetter(word.length()-1)) == false) {
2      wordlength--;
3  }
```

Un-selected is correct

```
1  if (Character.isLetter(word.charAt(word.length()-1)) == false) {
2      wordlength--;
3  }
```

Correct

This is a correct answer.

```
1  if ( ! Character.isLetter(word.charAt(word.length()-1))){
       wordlength--;
2  }
```

Correct

This is a correct answer.

```
if (word.charAt(word.length()-1) == false) {
    wordlength--;
}
```

Breaking the Caesar Cipher

11/11 points (100%)

Practice Quiz, 11 questions

Un-selected is correct

---

1 / 1 points

2.

Consider the following code which is in a program that counts the words of different lengths from a file, and assume **resource** is a **FileResource** to a file of words and **counts** is an array of integer counters.

```
for(String word : resource.words()){
    int wordlength = word.length();
    if (wordlength >= counts.length) {
        wordlength = counts.length - 1;
    }
    if (wordlength > 0 ) {
        counts[wordlength] ++;
    }
}
```

Which one of the following best describes the purpose of the first **if** statement?

- The **wordlength** of every word is reduced by one as the counter in the array is off by one since the first counter in the array is at index 0.
- All words that are of length greater than or equal to the size of the **counts** array are not counted in any counter.
- All words that are of length greater than or equal to the size of the **counts** array are counted in the last counter in the array.

Correct

This is the correct answer. If we didn't change the value of **wordlength** for words longer than the size of the **counts** array, our program would crash in the second **if** statement.

- All words that are of length greater than or equal to the size of the **counts** array are shortened to a smaller size that is countable before they are counted.

---

1 / 1 points

3.

Which one of the following words (or rather groups of characters with no blanks) might need a special case in the **countWordLengths** method?

- **Bird:,**

$

Breaking the Caesar Cipher

11/11 points (100%)

Practice Quiz, 11 questions

Correct

You must be careful you don't subtract it twice from the front and the end. This is a special case you must consider separately.

"Hello"

!-Check-!

---

1 / 1 points

4.

Consider the file **romeo.txt**.

What is the most common word length (ignoring the punctuation of the first and last character of each group of characters)?

4

Correct Response

---

1 / 1 points

5.

Consider the file **lotsOfWords.txt**.

What is the most common word length (ignoring the punctuation of the first and last character of each group of characters)?

8

Correct Response

---

1 / 1 points

6.

Consider the following code for the **halfOfString** method that is given a string named **message** and a starting position named **start**, and returns a new string that is formed from every other character starting with the given start location.

```
1  String answer = "";
2  // MISSING CODE
3  return answer;
```

Which one of the following is the missing code for this method?

- [x]
```
1  for (int k = start; k< message.length() ; k+= 2) {
2       answer = answer + message.charAt(k);
3  }
```

Correct
This is the correct answer.

- [ ]
```
1  for (int k = start; k< message.length() ; k+= 2) {          answer
       = answer + message.charAt(start);
2  }
```

- [ ]
```
1  for (int k = start+2; k< message.length() ; k+= 2) {          answer
       = answer + message.charAt(start);
2  }
```

- [ ]
```
1  for (int k = 0; k< message.length() ; k+= start) {
2       answer = answer + message.charAt(k);
3  }
```

- [ ]
```
1  for (int k = 0; k< message.length() ; k+= 2) {
2       answer = answer + message.charAt(start);
3  }
```


1 / 1 points

7.

Suppose we decided to write an eyeball method for the two-key decrypt algorithm.

Which one of the following obstacles would make using the eyeball method difficult?

- [ ] We would need to decrypt with all possible key combinations resulting in 26+26 = 52 possible combinations to look at.

- [x] We would need to decrypt with all possible key combinations resulting in 26*26 = 676 possible combinations to look at.

Correct

- [ ] We would need to decrypt with all possible key combinations resulting in 26+26 = 52 keys with 52*52 = 2704 possible key combinations to look at.

- [ ] We would need to decrypt with all possible key combinations resulting in 26+26 = 52 keys with 52+52 = 104 possible key combinations to look at.

Breaking the Caesar Cipher

11/11 points (100%)

Practice Quiz, 11 questions

1 / 1 points

8.

The following phrase was encrypted with the two key encryption method we discussed using the two keys 2 and 20.

Top ncmy qkff vi vguv vbg ycpx

What is the decrypted message?

(Note: You should preserve the spacing in your answer.)

Run like wild to beat the wind

Correct Response

1 / 1 points

9.

The following phrase was encrypted with the two key encryption method discussed in the programming exercise, using two unknown keys.

Akag tjw Xibhr awoa aoee xakex znxag xwko

What is the decrypted message?

(You'll need to figure out which keys were used to encrypt it.)

Eren and Emily have evil eerie green ears

Correct Response

1 / 1 points

10.

Decrypt the encrypted file **mysteryTwoKeysPractice.txt**

This file is encrypted with the two key encryption method we discussed. You will need to decrypt the complete file by figuring out which keys were used to encrypt it.

What are the first five decrypted words?

Geometric computing research at Duke

Breaking the Caesar Cipher

11/11 points (100%)

Practice Quiz, 11 questions

Correct Response

1 / 1 points

11.

Decrypt the encrypted file **mysteryTwoKeysPractice.txt**

This file is encrypted with the two key encryption method we discussed. You'll need to decrypt the complete file by figuring out which keys were used to encrypt it.

What are the two keys (in order) that were used to encrypt it?

(Note: Enter your answer as **firstkey**,**secondkey** with no spaces, for example:

3,15

17,4

Correct Response