Implementing the Caesar Cipher

Practice Quiz, 6 questions

1
point

1.

Which one of the following shows how you could assign the character variable **ch** to the lowercase version of itself, without having to write any additional methods?

○
```
Character.toLowerCase(ch);
```

○
```
ch = toLowerCase(ch);
```

○
```
ch = ch.toLowerCase();
```

○
```
ch = Character.toLowerCase(ch);
```

1
point

2.

Consider writing the method **isAorE** that has one char parameter, **ch**. This method should return true if **ch** is either ‘a’ or ‘e’, and otherwise return false.

Which two of the following are correct implementations for the method **isAorE**?

□
```
public boolean isAorE(char ch) {
     if (ch == 'ae') {
          return true;
     }
     return false;
}
```

□
```
public boolean isAorE (char ch) {
     if (ch != 'a' || ch != 'e') {
          return false;
     }
     return true;
}
```

□
```
public boolean isAorE (char ch) {
     if (ch == 'a' || ch == 'e') {
          return true;
     }
     return false;
}
```

□

□

```
public boolean isAorE (char ch) {
    if (ch == 'a') {
        return true;
    }
    else {
        return false;
    }
    if (ch == 'e') {
        return true;
    }
    else {
        return false;
    }
}
```

□

```
public boolean isAorE (char ch) {
    if (ch == 'a') {
        return true;
    }
    if (ch == 'e') {
        return true;
    }
    return false;
}
```

1 point

3.

Assume that **isVowel** is a method with one char parameter, and that this method returns true if that character is a vowel, and false if it is not a vowel.

Consider the following code to replace all vowels in a string **phrase** with a given character **ch**.

```
StringBuilder sb = new StringBuilder(phrase);
for (int k=0; k < sb.length(); k++) {
    // MISSING CODE
}
```

Which one of the following is the missing code to modify **sb** so that all vowels in **sb** are replaced with the character **ch**?

○

```
if (isVowel(sb[k] == ch)) {
    sb[k] = ch;
}
```

○

```
if ( (isVowel(sb[k]))) {
    sb[k] = ch;
}
```

○

```
if (isVowel(sb.charAt(k))) {
    sb.setCharAt(k, ch);
}
```

○

```
if ( (isVowel(sb[k])) == ch) {
    sb[k] = ch;
}
```

○

```
if (sb.charAt(isVowel(k))) {
    sb.setCharAt(k, ch);
}
```

○

```
if (isVowel(sb.charAt(k)) == ch) {
    sb.setCharAt(k, ch);
}
```

1
point

Implementing the Caesar Cipher

Practice Quiz, 6 questions

4.

Consider the following definition where **phrase** is a String.

```
StringBuilder sb = new StringBuilder(phrase);
```

Which <u>two</u> of the following are correct ways to ask if a character in the kth position of **sb** is equal to the character **ch**, when we want them to match regardless of case?

□

```
if (Character.toLowerCase(sb.charAt(k)) == Character.toLowerCase(ch))
    {
```

□

```
if (Character.toLowerCase(sb[k] == ch)) {
```

□

```
char one = Character.toLowerCase(sb.charAt(k));
char two = Character.toLowerCase(ch);
if (one == two) {
```

□

```
if (sb.charAt(k).toLowerCase() == Character.toLowerCase(ch)) {
```

□

```
char one = sb.charAt(k).toLowerCase();
char two = Character.toLowerCase(ch);
if (one == two) {
```

1
point

5.

Encrypt the following phrase with Caesar Cipher key 15.

At noon be in the conference room with your hat on for a surprise party. YELL LOUD!

What is the encrypted string?

(Note: Your encrypted string should preserve spacing and punctuation.)

Enter answer here

1
point

6.

Encrypt the following phrase with the algorithm described for using two Caesar Cipher keys, with key1 = 8 and key2 = 21.

At noon be in the conference room with your hat on for a surprise party. YELL LOUD!

What is the encrypted string?

Implementing the Caesar Cipher

Practice Quiz, 6 questions

(Note: Your encrypted string should preserve spacing and punctuation.)

Enter answer here

Submit Quiz