1. Fill in the blanks such that the sequence 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 is printed.

```
for (___a___; ___b___; ___c___) {
    System.out.println(i);
}
```

2. Complete the following truth table. Feel free to write intermediate steps in the blank space.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>!(a &amp;&amp; b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>(a)</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>(b)</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>(c)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>(d)</td>
</tr>
</tbody>
</table>

3. Rewrite the following code to be simpler.

```
if (x < 0) {
    return true;
} else if (x <= 100) {
    return false;
} else {
    return true;
}
```

4. Rewrite the following code to be simpler.

```
if (s.length() % 2 == 0) {
    return false;
} else {
    return true;
}
```

5. What is the output of the following code?

```java
String s = "pul";
for (int i = 0; i < s.length(); ++i) {
    System.out.print(s.charAt(i));
}
for (int i = 0; i < s.length(); ++i) {
    System.out.print(s.charAt(s.length() - 1 - i));
}
```

6. Rewrite the following expression to be simpler.

```
a || (!a && b)
```
7. Enumerate the contents of the array `numbers` after the following code is executed.

```
int[] numbers = new int[5];
numbers[0] = 3;
for (int i = 1; i < numbers.length; ++i) {
    numbers[i] = numbers[i - 1] * 2 + 1;
}
```

8. Rewrite this `while` loop as a `for` loop.

```
char c = 'a';
while (c <= 'z') {
    System.out.println(c);
    c++;
}
```

9. Rewrite the following code to be simpler.

```
if (a.length == 0) {
} else {
    System.out.println(a[0]);
}
```

10. Fill in the blank in the following code so it doesn’t throw an index out-of-bounds exception.

```
public static boolean firstIsLast(int[] counts) {
    return ______________counts[0] == counts[counts.length - 1];
}
```

11. Declare and initialize a new `ArrayList` named `friends` that holds `Strings`.

12. Write a single statement that initializes an array named `choices` to hold the words `rock`, `paper`, and `scissors`.

13. Write method `addN` to accept three parameters: an `ArrayList` of `String`, an item of type `String`, and a quantity of type `int`. It appends the item to the list the specified number of times.

14. Write method `zeroify` to accept an array of `int`. It alters the array such that any negative element is set to 0.

15. Write method `grid` to accept two parameters: a width and a height, both of type `int`. It prints an alternating pattern of forward and backward slashes of the specified resolution. For example, `grid(7, 4)` produces this output:

```
/\////////
/\////////
/\////////
/\////////
```