Lab 2, Tasks 1 and 4

Task 1

1) line of code | a | t | s
--- | --- | --- | ---
`s = 'cs111'` | 'cs111' | | |
`t = 'is amazing!'` | 'cs111' | 'is amazing!' | |
u = s[1] + t[-4:] | 'cs111' | 'is amazing!' | 'sing!' |
s = s[ :2] + (t[-1]*2) | 'cs!!' | 'is amazing!' | 'sing!' |
t = t[1:-2:2] | 'cs!!' | 'saai' | 'sing!' |
s[::-2] | 'cs!!' | 'saai' | 'sing!' |

2) output:
cs!! saai sing!

6) To see the effect of the skip-slicing line (the final line in the table), change that line to:

`s = s[:::-2]`
Task 4

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Notes:
- c // 2 performs `integer` division, so we get 6 and not 6.5
- We don't execute the block associated with the second `elif`, even though b == 3 is true. That's because the if-elif-elif-else is a single four-way decision, and we never execute more than one block in a multi-way decision.

Because the initial if condition is true, we execute its block and we don't even look at the `elif` conditions. Rather, we skip to the statement that comes after the multi-way decision, which in this case is the second if statement.