

Invalid AOML string:

```
<font color=green><font color=red>888<font  
color=blue>999<center>123<right>3</font>4</font>5</div>6</center>678
```

Expected html output:

888999
123
345
6
678

(Not exactly like this in AO due to the lack of <right> and </div> being a valid closing tag for <center>, but I will not cover those cases)

Resolve missing tags:

- For each OpenNode, add its name to the top of the stack
- For each CloseNode, remove the topmost entry with the same name from the stack
 - If no entry was found in the previous step, add an OpenNode before CloseNode
- Append a CloseNode for each entry that remain on the stack (start with the top of the stack)

```
<font color=green><font color=red>888<font  
color=blue>999<center>123<right>3</font>4</font>5<div></div>6</center>678
```

```
<font color=green><font color=red>888<font  
color=blue>999<center>123<right>3</font>4</font>5<div></div>6</center>678</right>
```

```
<font color=green><font color=red>888<font  
color=blue>999<center>123<right>3</font>4</font>5<div></div>6</center>678</right></font>
```

Resolve elements:

- For each OpenNode, add its name to the top of the stack
- For each CloseNode, check if its name is on top of the stack
 - If it is, remove it and continue
 - If it's not and the element is an inline element:
 - Prefix and postfix a CloseNode and OpenNode to the top element on the stack
 - Iterate down the stack with this same process till the right OpenNode is found
 - Remove the matching OpenNode from the stack
 - If it's not and the element is a block element:
 - Prefix and postfix a CloseNode and OpenNode around the offending CloseNode
 - Iterate down the stack with this same process till the right OpenNode is found
 - Remove the matching OpenNode from the stack

Stack: div, font, font, **font**, center, **right**

```
<font color=green><font color=red>888<font color=blue>999<center>123</font><right><font color=blue>3</font>4</font>5<div></div>6</center>678</right></font>
```

Stack: div, font, font, **font**, **center**, right

```
<font color=green><font color=red>888<font color=blue>999</font><center><font color=blue>123</font><right><font color=blue>3</font>4</font>5<div></div>6</center>678</right></font>
```

Stack: div, font, **font**, **center**, **right** (doing 2 iterations at once here)

```
<font color=green><font color=red>888<font color=blue>999</font></font><center><font color=red><font color=blue>123</font></font><right><font color=red><font color=blue>3</font>4</font>5<div></div>6</center>678</right></font>
```

Stack: div, font, font, **center**, **right**

```
<font color=green><font color=red>888<font color=blue>999</font></font><center><font color=red><font color=blue>123</font></font><right><font color=red><font color=blue>3</font>4</font>5<div></div>6</right></center><right>678</right></font>
```

Ouput:

888999
123
345
6
678