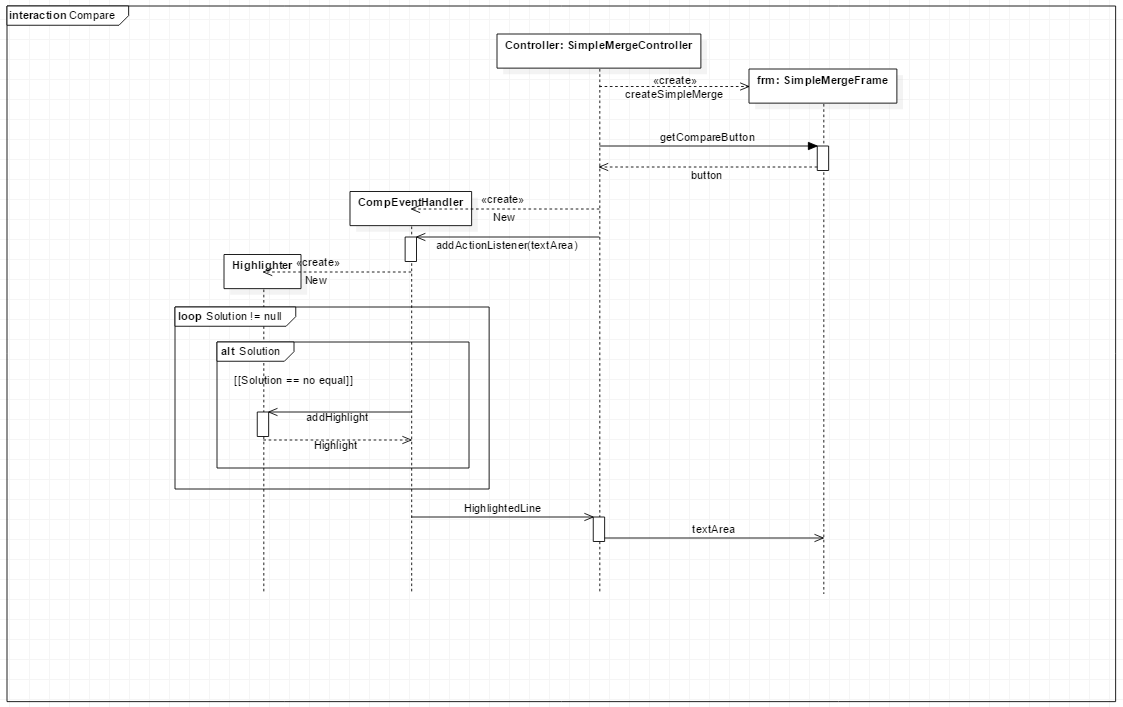
3. Save

4. Compare

1. System Sequence Diagram

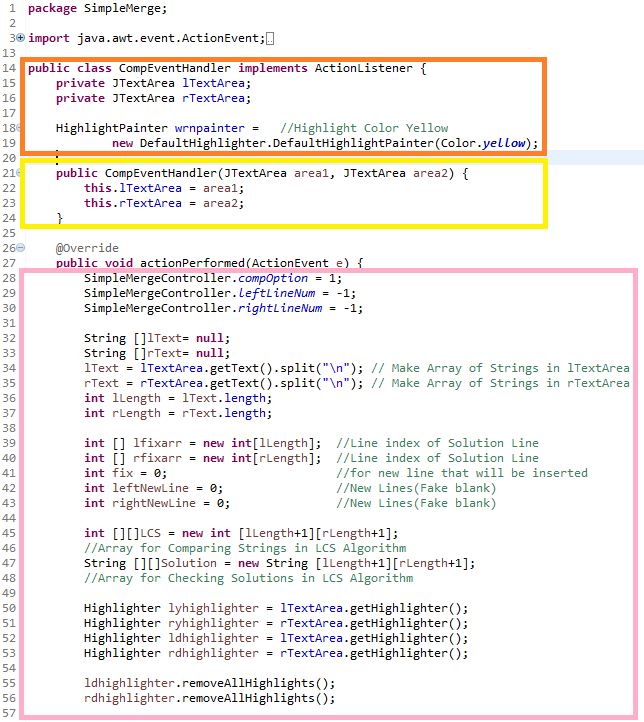


Description

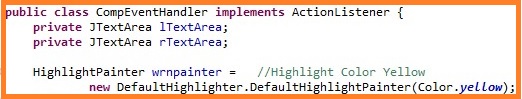
Premise) Main Class’s Function creates Controller Instance.

1. The user clicks the compare button.
2. LCS algorithm is applied
3. Get yellow highlights from highlighter.
4. Highlight different lines.

B) Implementation



1) Dependency of CompEventHandler

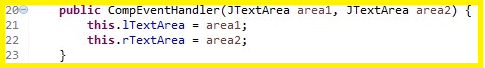


CompEventHandler implements the ActionListener interface.



Structural form is identical for all sub-controller classes (handlers).

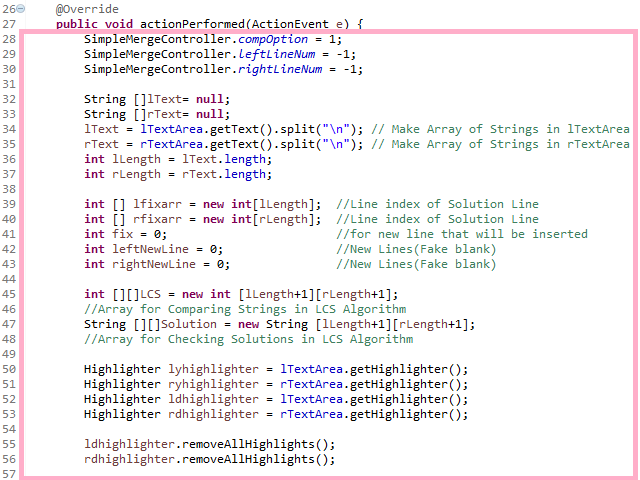
2) TextArea’s Role in CompEventHandler



CompEventHandler acts as a sub-controller for Compare case for the Controller instance.

So, CompEventHandler class receives access to two textareas.

3) Declarations



Set the compOption 1. It makes users enable to do copy to left, copy to right.

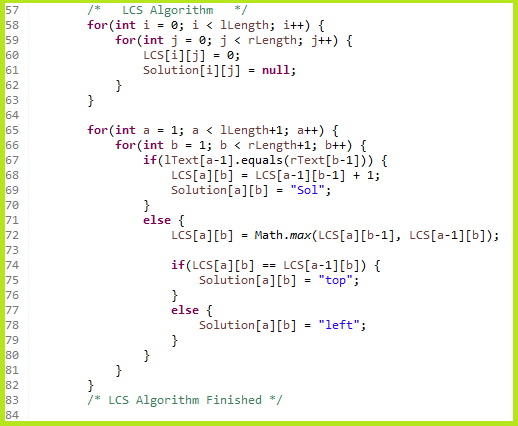
We must compare each lines in leftTextArea and rightTextArea.

First, split each lines in TextAreas and save them in lText array and rText array.

Then, we will align same Strings in same line by using lcs algorithm, and highlight the different lines.

So, we should declare arrays for LCS algorithm, arrays for inserting new Lines, and highlighters.

4) LCS Algorithm



LCS algorithm means the Longest Common Subsequence algorithm. We use two String arrays in this algorithm, and it will find the state that matches same Strings as many as possible.

“Sol” means lText[a] and rText[b] are same. “top” and “left” means lText[a] and rText[b] are different.

5) Highlight different Lines



Solution Array contains “Sol”, “top”, “left”.

If Solution array has “Sol”, it means lLength’s text and rLength’s text are equal. We want to align these lines in same line, so save each line’s index in rfixarr array and lfixarr array.

If Solution array has “top” or “left”, it means they have different lines. When we align “Sol” lines,(it means we should insert new line “\n”), another text line that new line is not inserted should be highlighted. So, get make lyhighlighter and ryhighlighter to highlight those lines.



We made rLength--, lLength-- above, so when rLength or lLength become 0, and index of that array is Sol, rLength or lLength will be -1 and it is array out of bounds exception. So, I made two conditions.

Now, we can consider all lines of both textareas.

There are rstart, rend, lstart, and lend. These variables contains index of position in each textarea. For example, If lTextarea’s first line is “abc”, and second line is “ef”, lstart will be 4. Later, I made these variables because highlighter.addHighlight method use values of this type.

6) Add new Lines(fake blank)

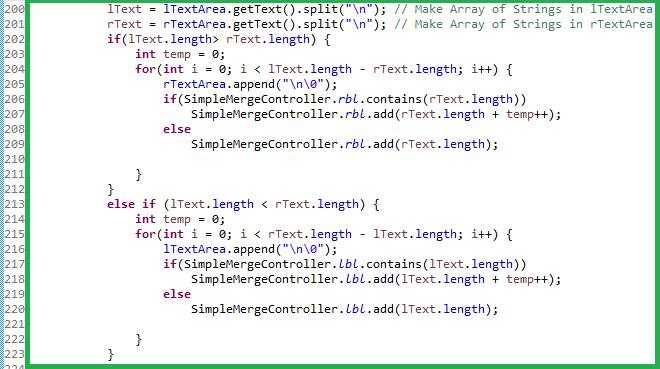


We should align same strings in same line. By using lfixarr and rfixarr arrays we saved indexes while highlighting, insert new line “\n” in appropriate space and save line index in rbl and lbl.

If new line (“\n”) is inserted, index of each line will be altered. That is why I made leftNewLine and rightNewLine variable. It makes correct position whenever “\n” is inserted.

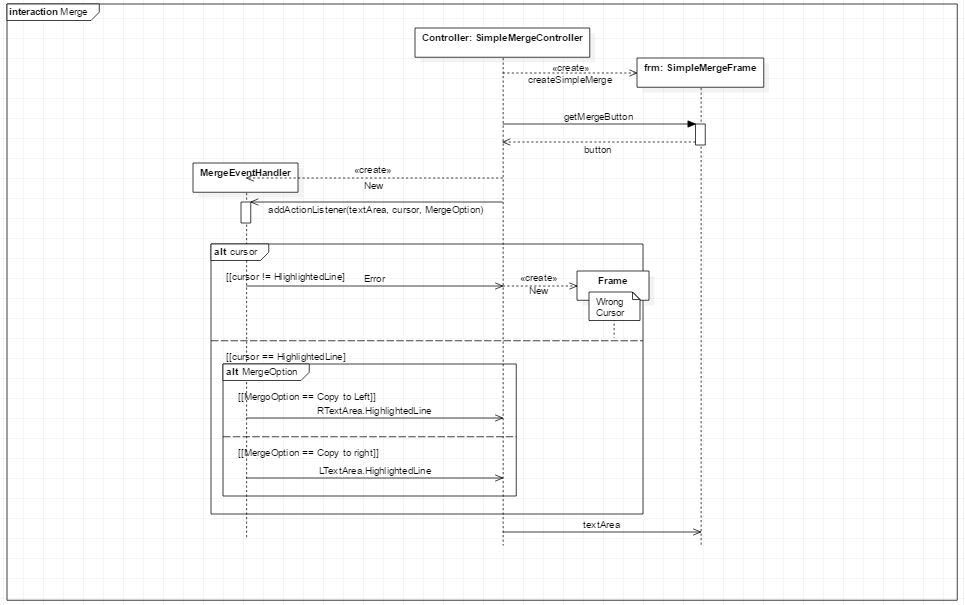
rbl and lbl are the SimpleMergeController’s static variable and it will be used in SaveEventHandler to erase fake blank when save contents of each textarea.

7) Add extra new Lines(fake blank)



We wanted to make two textareas have same lines, so insert temporal strings in shorter textarea. These lines are also added in lbl or rbl and will be deleted when user clicks the save button.

1. Merge
2. System Sequence Diagram



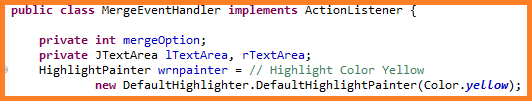
Description

Premise) Main class’s Function creates Controller Instance.

1. The user clicks copy to right or copy to left button.
2. Controller gives textArea, cursor, mergeOption to MergeEventHandler.
3. If cursor is not at highlighted line, return error message
4. If cursor is at highlighted line, CompEventHandler copies that lines to same line of another textarea.

B. Implementation

1) Dependency of MergeEvenHandler

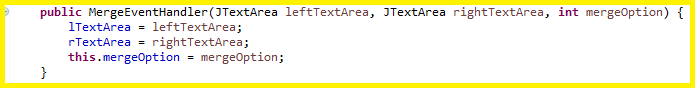


MergeEventHandler implements the ActionListener interface.



Structural form is identical for all sub-controller classes (handlers).

2) TextArea’s Role in MergeEventHandler



CompEventHandler acts as a sub-controller for Compare case for the Controller instance.

So, CompEventHandler class receives access to two textareas.

We also have mergeOption as a argument..

mergeOption 1 means copy to right, and 2 means copy to left.

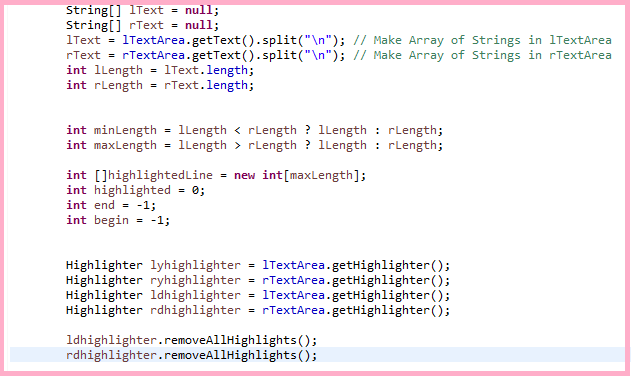
3) Preceding Condition



If user clicks copy to left or copy to right button before the compare button,

mergeEventHandler will not work.

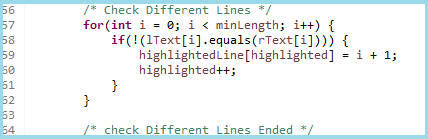
4) Declaration



Now, we already applied LCS algorithm, so we don’t need any algorithm anymore.

Just needs arrays of strings in each textarea, values to check for whether the highlighted lines are contiguous(end, begin), and highlighters.

5) Check Different Lines

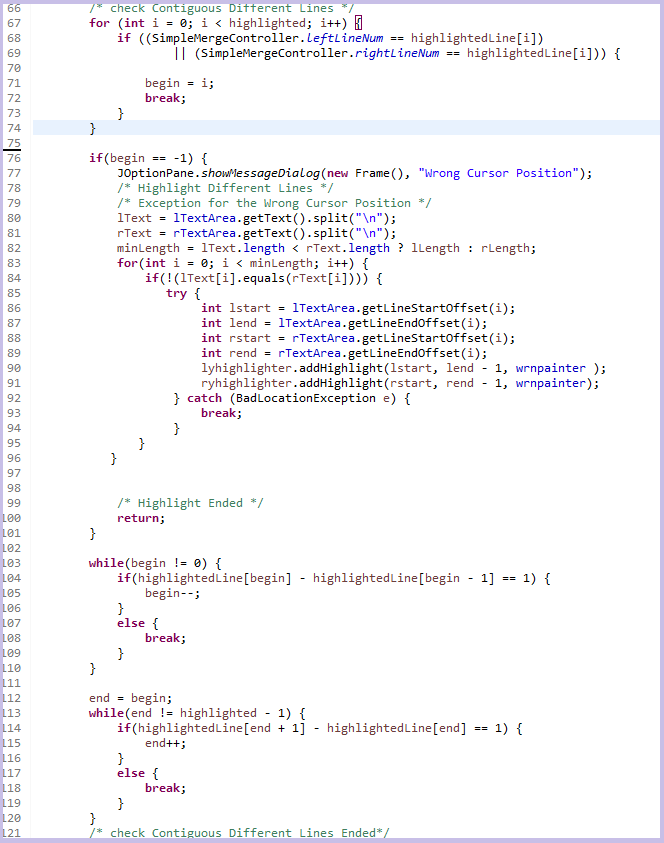


Lines are well aligned in each textarea when compare is completed.

So just check different lines.

highlightedLine array’s index is the different line number, and it will be used to check whether highlighted lines are contiguous or to highlight different lines again.

6) Check Contiguous Different Lines



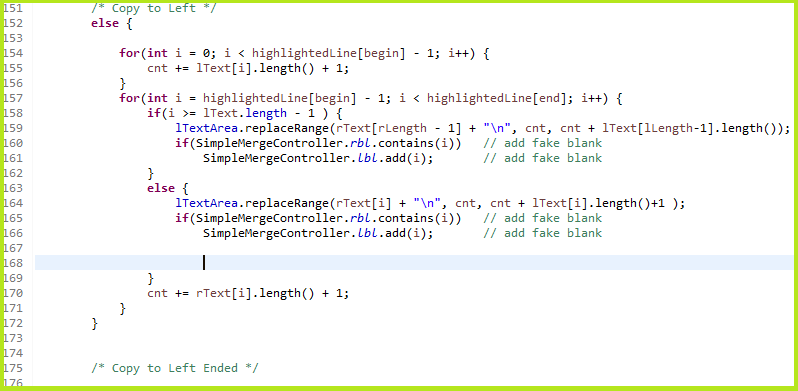
The value begin means the current cursor position.

If user didn’t click the textarea, the default value is -1.

So it will show “Wrong Cursor Position” when begin is -1. Also, because we delete all highlights when mergeEventHandler is started and wrong cursor window will end the program, highlight the different lines again.

Else, check if current line is the part of contiguous different lines. The highlighted line block will be copied together like winmerge program.

7) Copy Lines



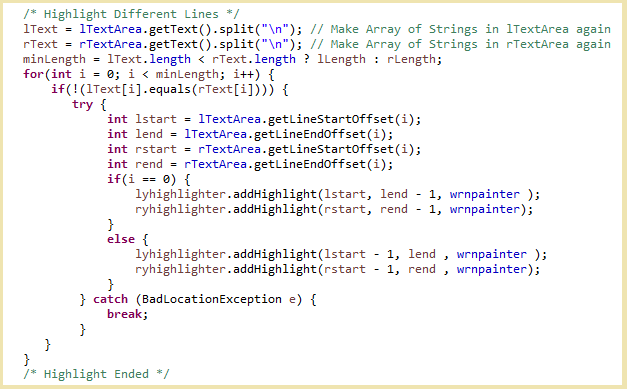


Copy different Line to left or right. Because TextArea.replaceRange method use the index of position in each textarea, make c variable to copy different line at the same time.

If the original string is fake blank(We saved the fake blank line in lbl or rbl),

We should also copy the fake blank then save this line in lbl or rbl.

7) Highlight Different Lines



Finally, highlight different lines again after copy the lines to left or right.

It works as CompEvenHandler’s highlighting part.

6. Controller

7, Viewer

III. Program Testing

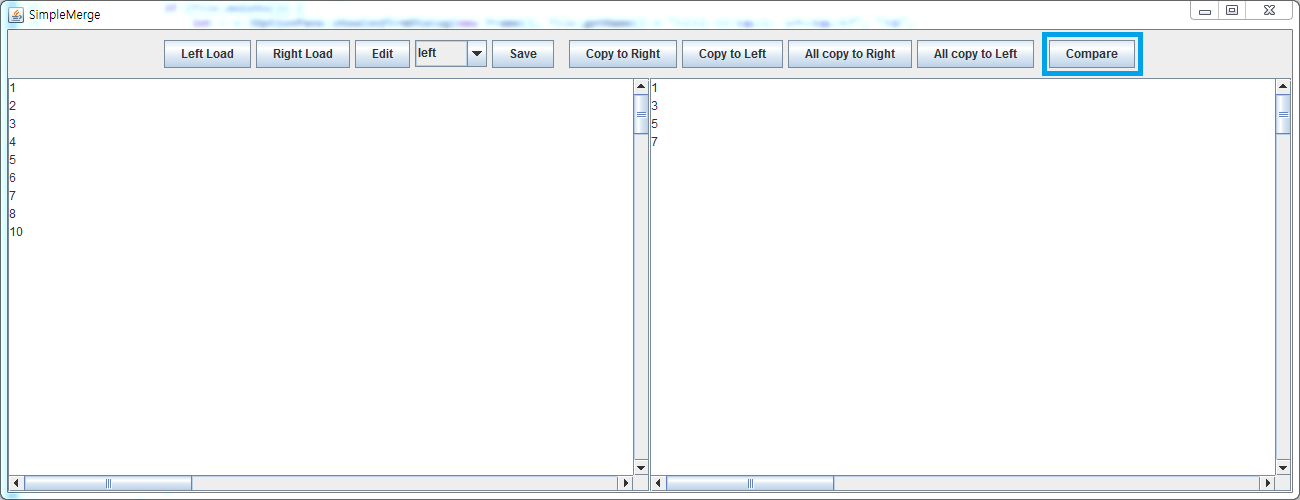
IV. Results

1.

2.

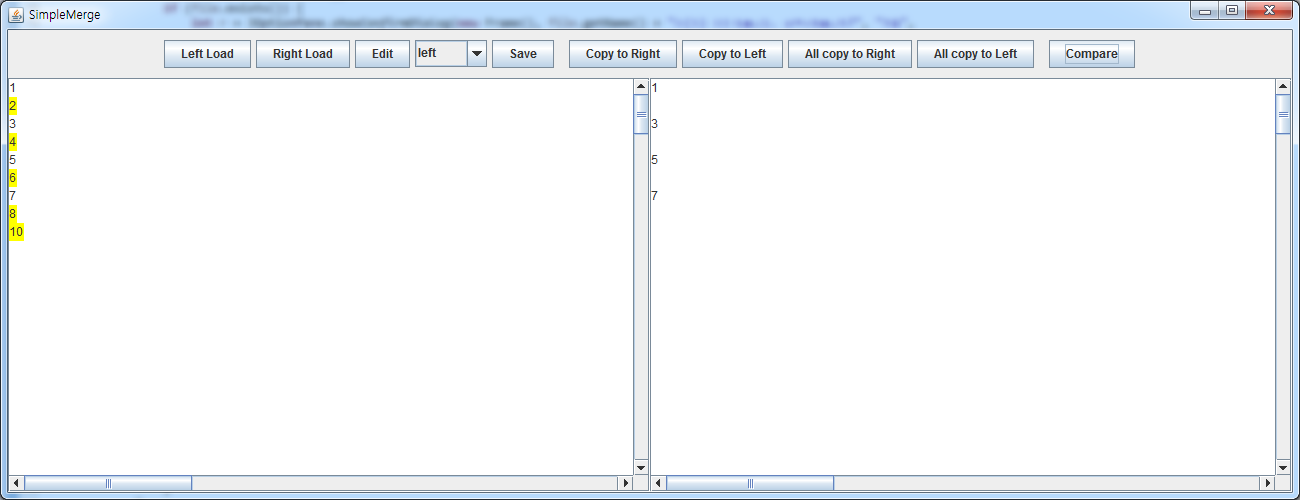
3.

4. Compare

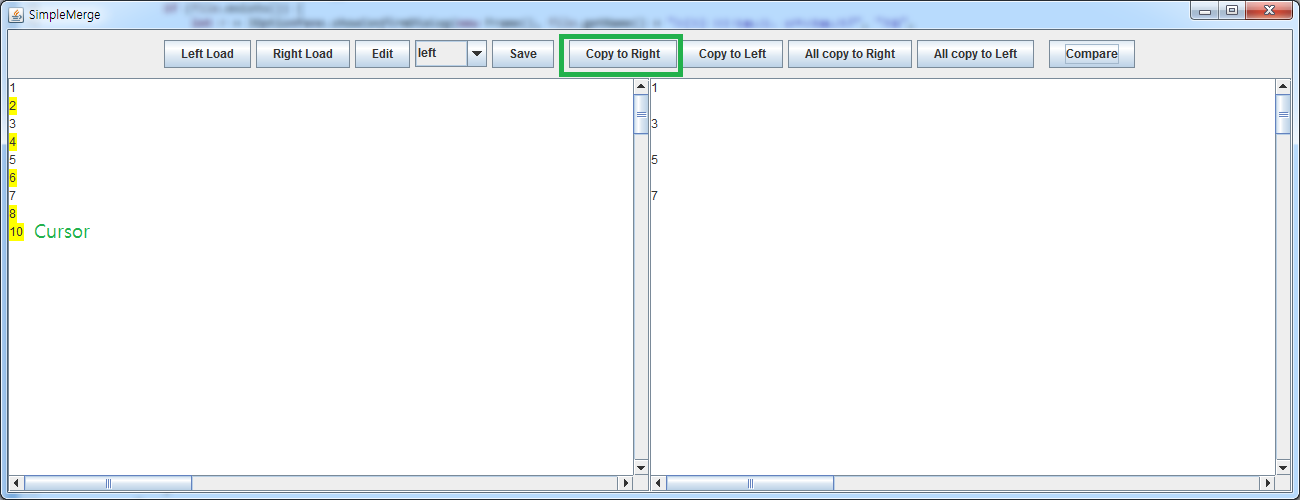


Press Compare Button

- When you press the compare button, LCS algorithm will be applied and Lines will be aligned.

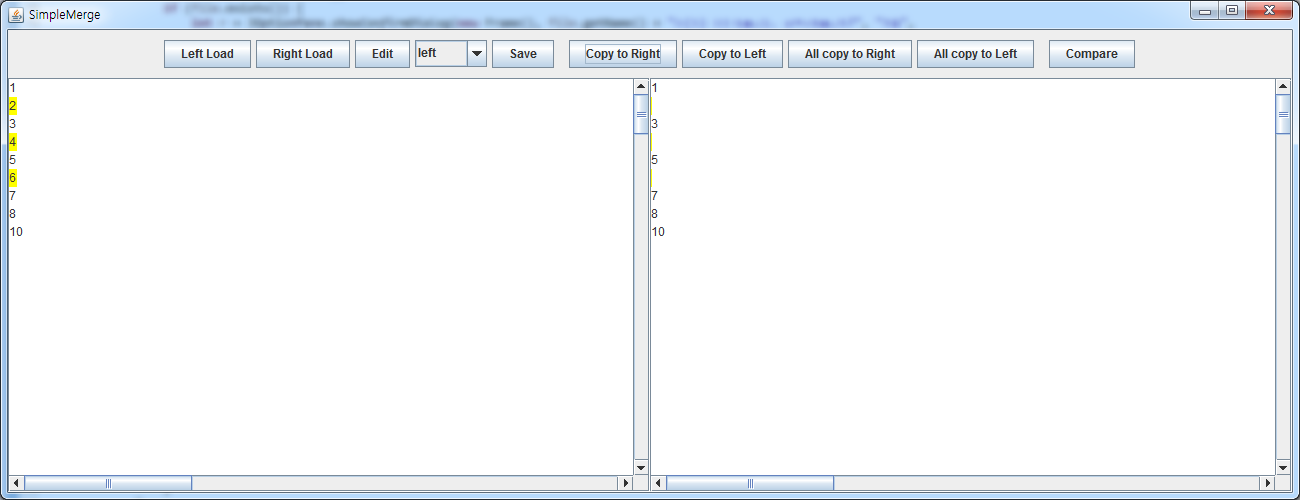
Result-

5. Merge



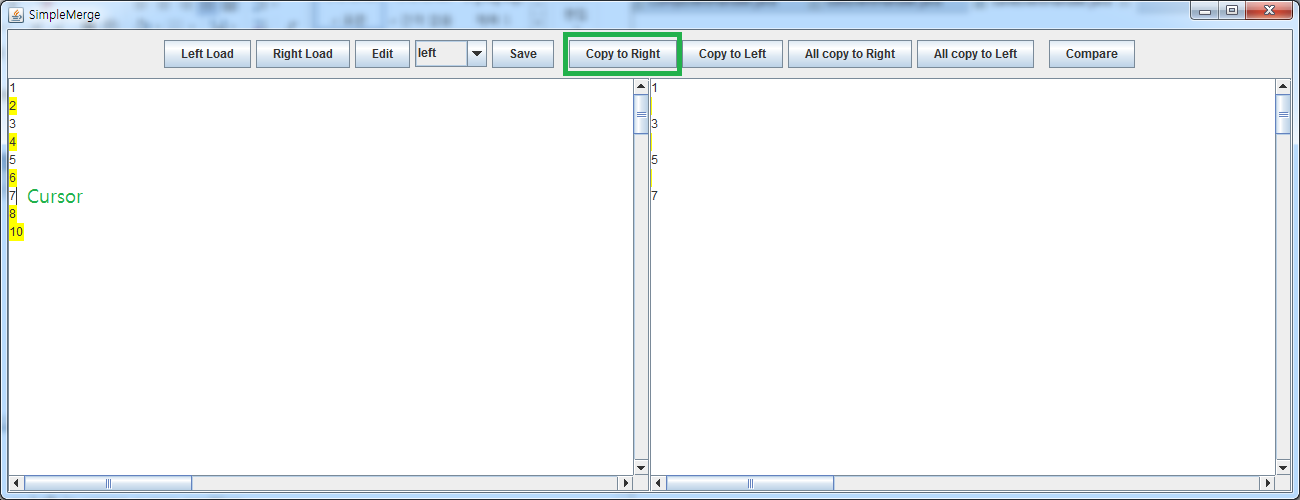
1) Press Copy to Right or Left Button in correct cursor position

- When you press the Copy to ~ button in a highlighted line, content of the line will be copied to another textarea.



2) Press Copy to Right or Left Button in wrong cursor position

- When you press the Copy to ~ button in a non-highlighted line, warning window will pop.



Result -

