Analysis & Design, Implementation, and Test

Document

개체이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

TEAM 16

**김은경 (20163927) 김소연 (20165417) 김수진 (20160342) 남유선 (20163228)**

**손승표 (20162581) 윤신영 (20163657)**

1. **Domain model (class diagram)**

텍스트이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

(you can see more clear diagram images at Github ‘Diagram’ folder!)

1. **Software architecture & Design model**

텍스트이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

텍스트, 지도이(가) 표시된 사진

높은 신뢰도로 생성된 설명[System sequence diagram]

1. **Major design decision**

// 수진, 유선 part

[How to compare the texts?]

By using the algorithms for longest common subsequence(LCS) problem and sequence alignment, we compute the matrix C which stores the LCS length and compare two panels based on the matrix.

[How to distinguish two cases? 1. Different strings are in same line index in each panel 2. The string is in only one panel]

For case 1, we multiply (-1) to the index of string. For case 2, the string is stored in the panel which the string is in and the blank value(0) is stored in another panel. This is for computing the start and end indices of blocks. When the getter function is called, returned data is the arraylist that all elements of left panel are changed to negative integer and all elements of right panel are changed to positive integer to distinguish two panel’s indices.

[What data type would be returned to Merge class after comparing in FileComparator class?]

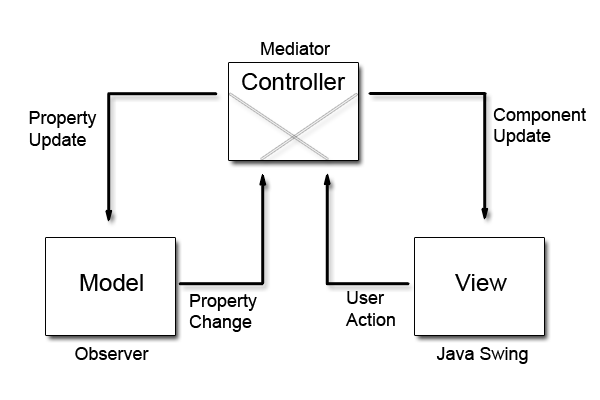
In FileComparator class, the difference of two panel is stored in each integer type arraylist and the block, which store the block’s start and end index, is stored in integer array type arraylist. When merge call getter functions by FileComparator object, all elements in left panel’s arraylists are changed to negative integer, and all in right panel’s are changed to positive integer.

[What data type would be returned after calling traverse function?]

There is a cursor for traverse function which is an integer type index for the block array and it represents current location of cursor. If traverse function is called, the block array and the cursor are returned.

1. **Explanation about how MVC concept was applied**

Here is the picture of MVC model. When we design this program, we referred to this picture. The point is that ‘Model’ can’t be directly connected to ‘View’.



We define ‘view’ classes as {MainView, PanelView, TextLineNumber, CompareTable, CompareTableRenderer}, ‘model’ classes as {Merge, TextEditorModel, FileComparator}, and ‘controller’ as {MergeController, TextEditorController}.

As you can see these two class diagrams (domain model and design model), all the classes in view are not directly connected to any class in model. In other words, controller is the only mediator between model and view. Also, when state of model is changed, controller receives the signal and sends it to the view so that the view can apply the change and show it to the user.

So, concept of MVC model is applied to our program.

1. **Explanation about how OO design principles were applied (with code examples)**

[Encapsulation]

We set member variables as private or protected (there are only few exceptions), also we made public getter & setter methods to access those private or protected member variables. For some methods which are used only within in one class, we set them as private, otherwise, we set methods as private.

[Inheritance]

We use inheritance when we design ‘view’ classes. Here are the pictures of code.

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1. **Explanation about how our program was designed to be testable by Unit-test tools**
2. **Usage of program & screen shots of examples**

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1. 프로그램의 시작화면. 활성화되어 있는 Load 버튼을 클릭하여 파일 load

스크린샷, 컴퓨터이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

1-1. load한 파일이 중복인 경우 중복임을 알리는 message dialog 출력

스크린샷, 컴퓨터, 실내이(가) 표시된 사진

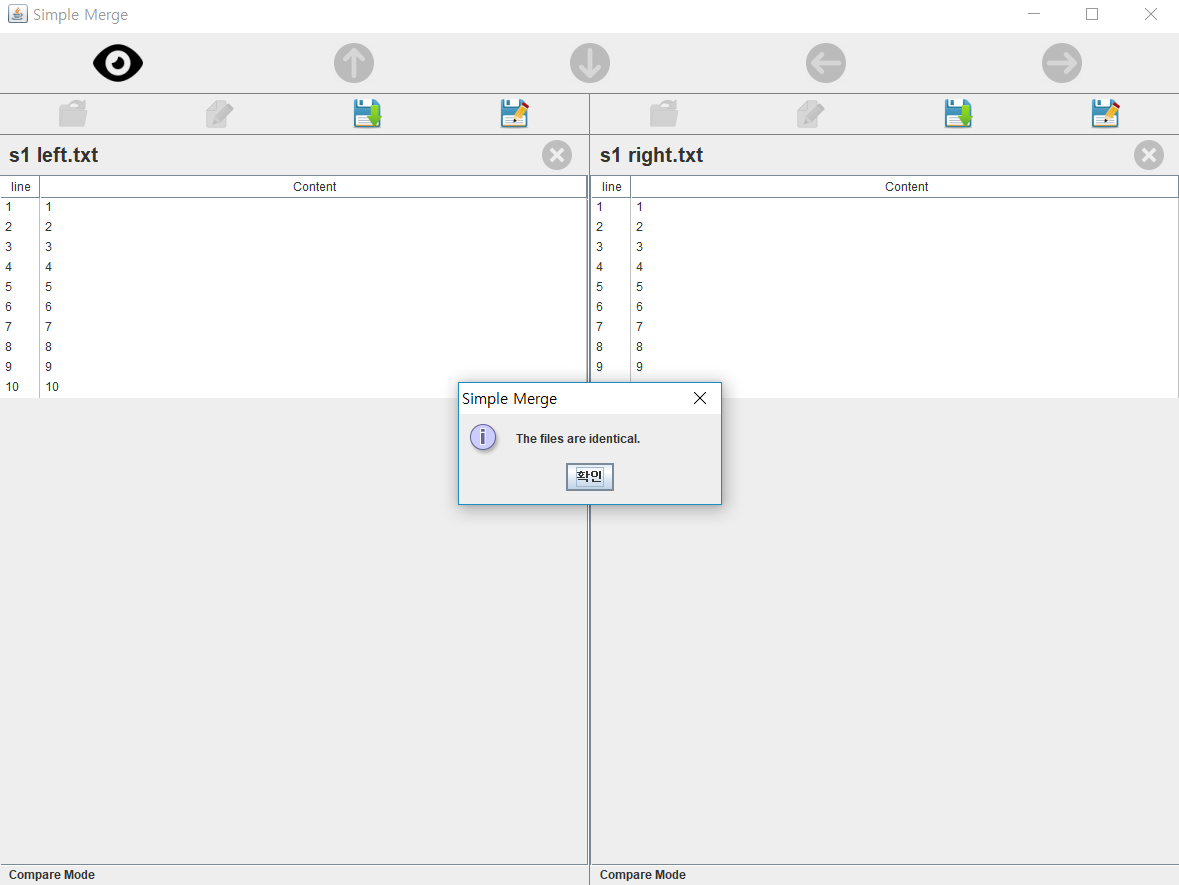
매우 높은 신뢰도로 생성된 설명

2. 파일이 Load된 상태. 패널의 밑부분에는 현재 패널의 모드가 표시되고(스크린 샷은 View Mode) 현재 커서가 있는 줄의 index는 빨간색으로 표시됨.

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

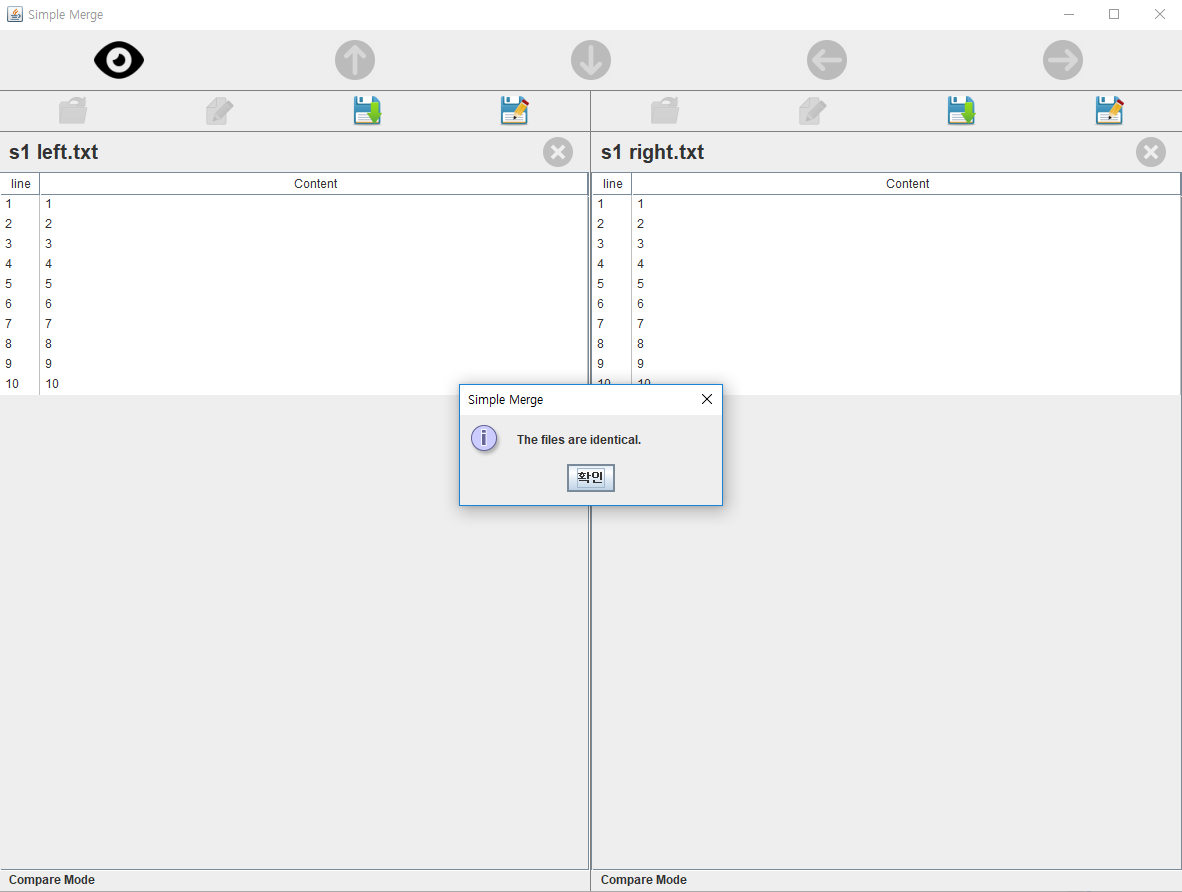
3. compare 버튼을 클릭한 후, Traverse와 Merge 버튼이 활성화됨. 줄 정렬을 위해 추가된 공백줄은 회색으로 표시되고, index는 ‘-‘로 표시됨. 더 이상 traverse할 수 있는 블록이 없으면 해당 버튼 비활성화.



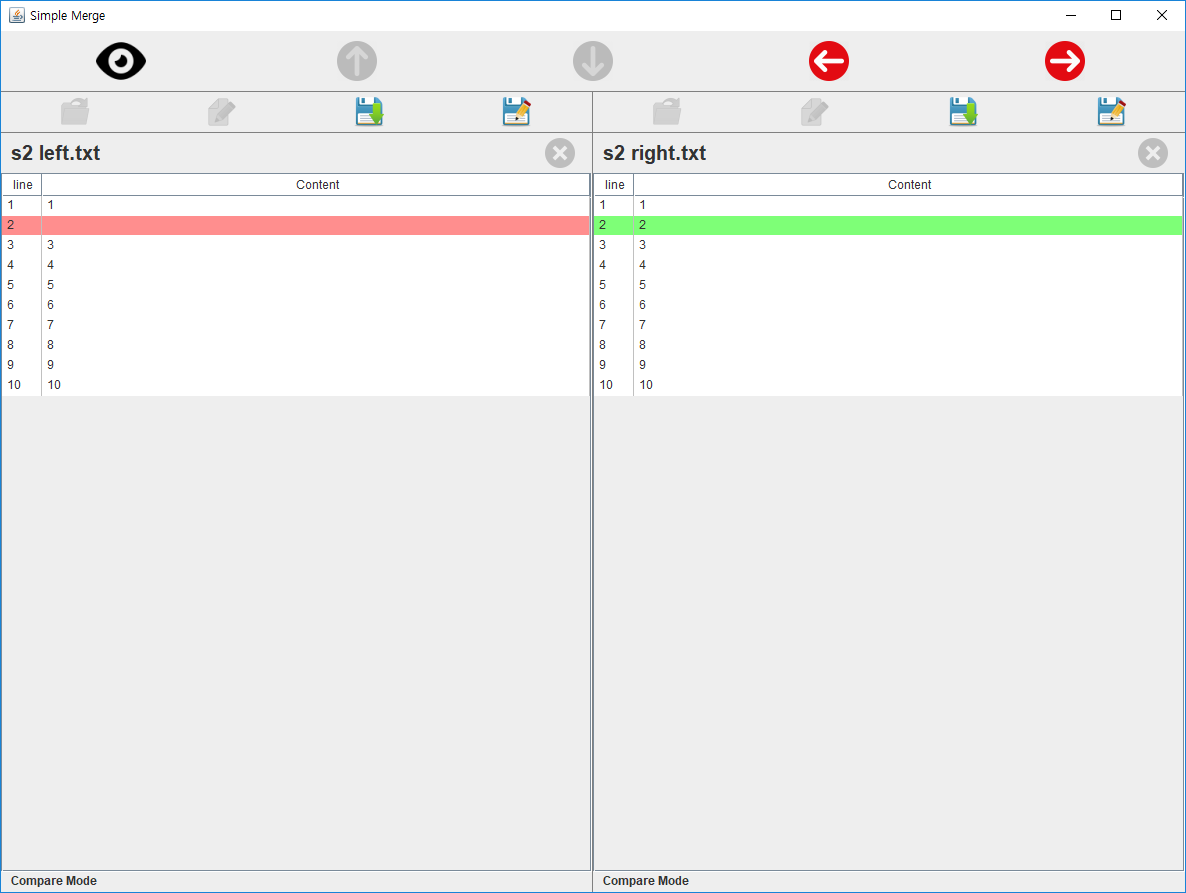
3-1. compare 연산 결과 두 텍스트파일이 동일한 경우 이를 알리는 message dialog 출력

1. **Functional unit test cases and results**
2. **System test cases and results**

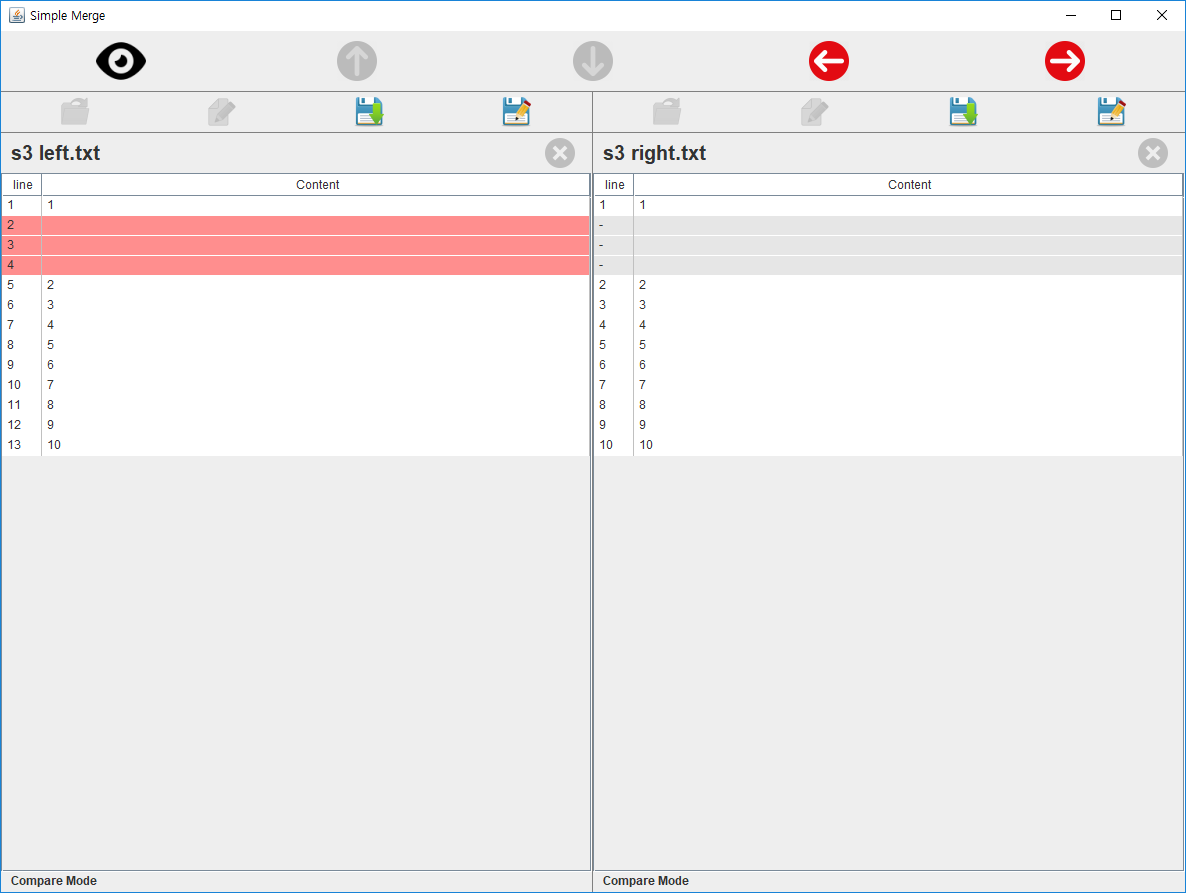
**S1. Result :** 두 파일이 동일함을 보여준다.



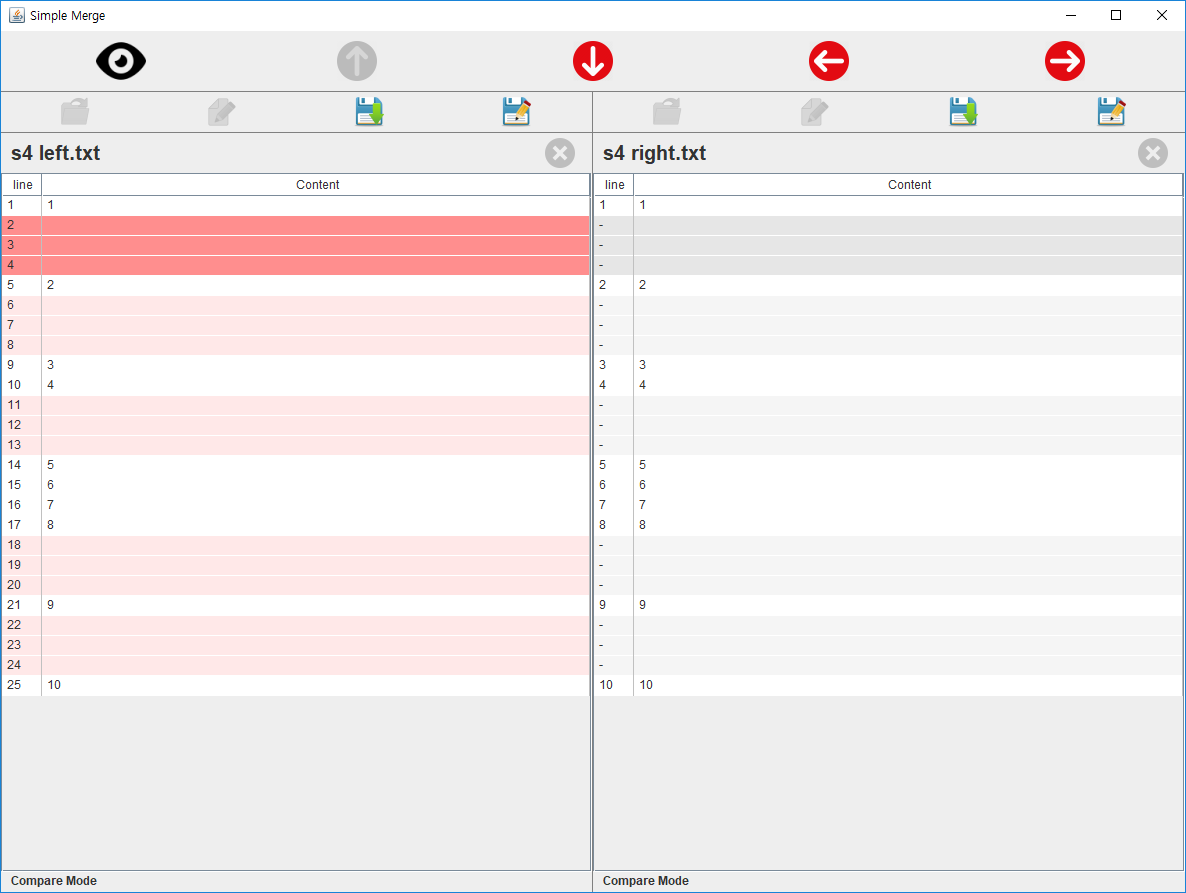
**S2. Result :** 왼쪽 파일에 2가 없음이 나타난다.



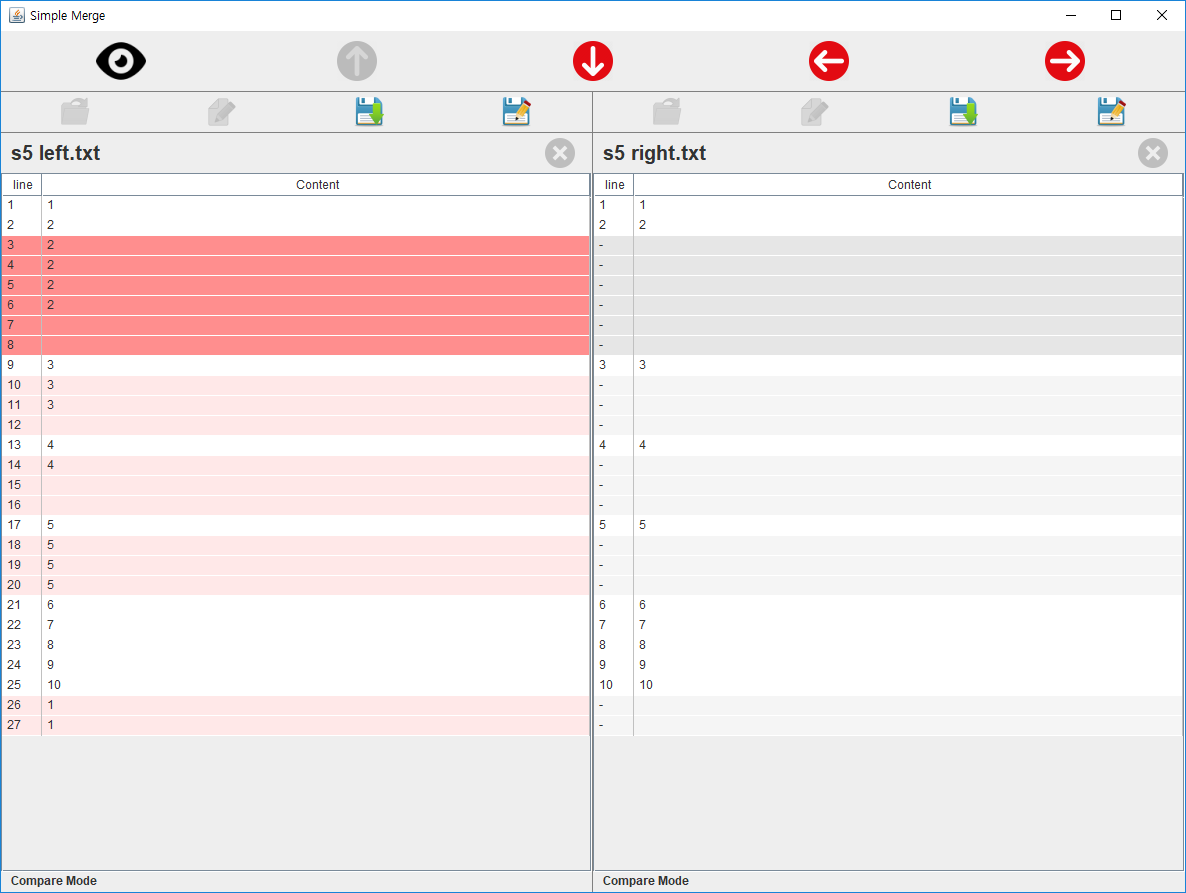
**S3. Result :** 왼쪽 파일에 빈줄이 추가되어 있음이 나타난다.



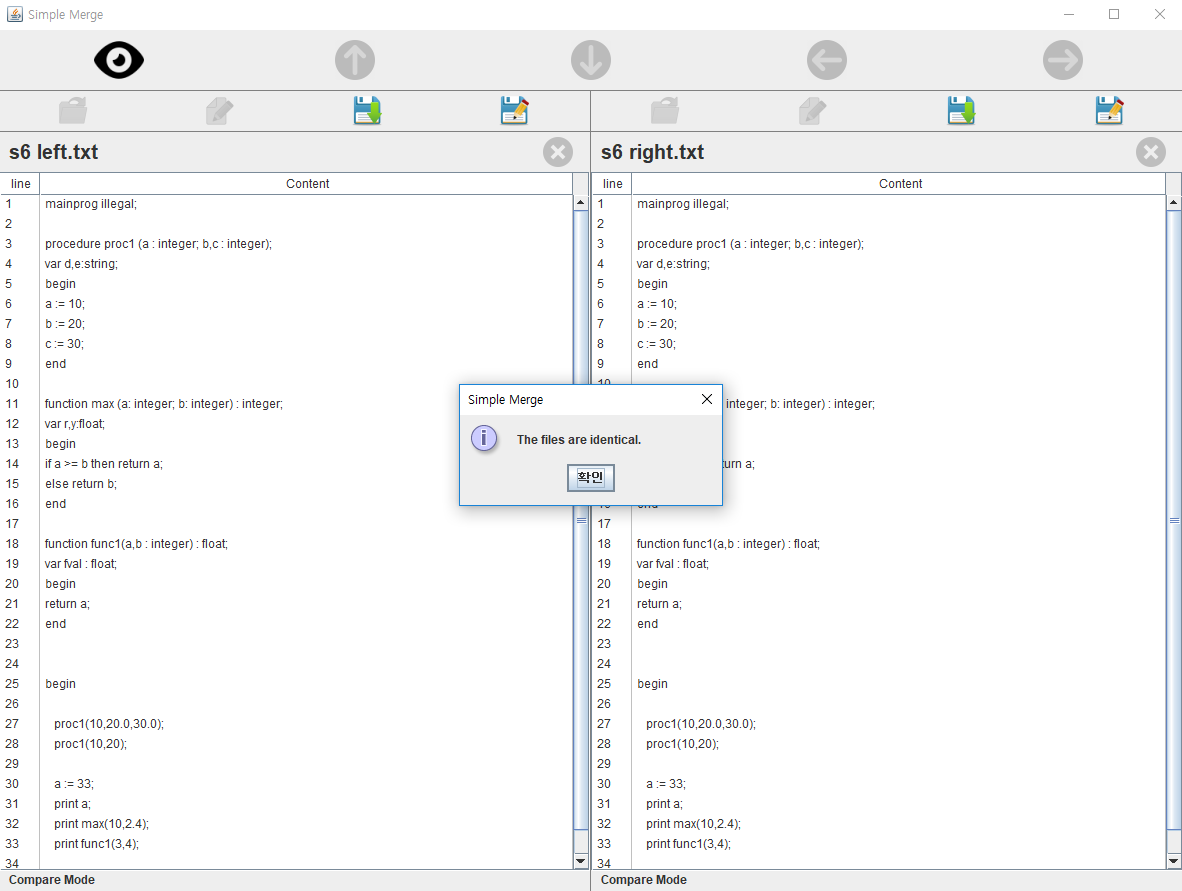
**S4. Result :** 여러 개의 빈 줄 블록이 있음이 나타난다.



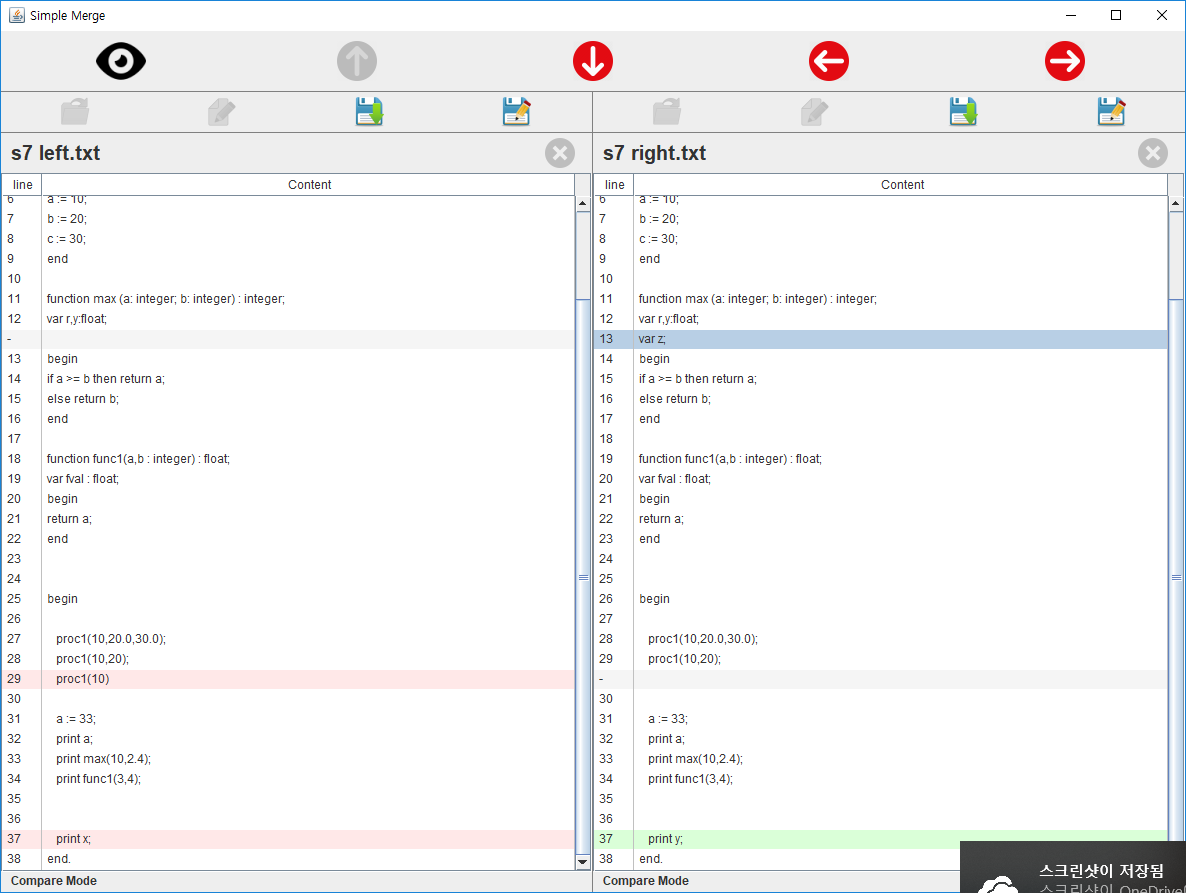
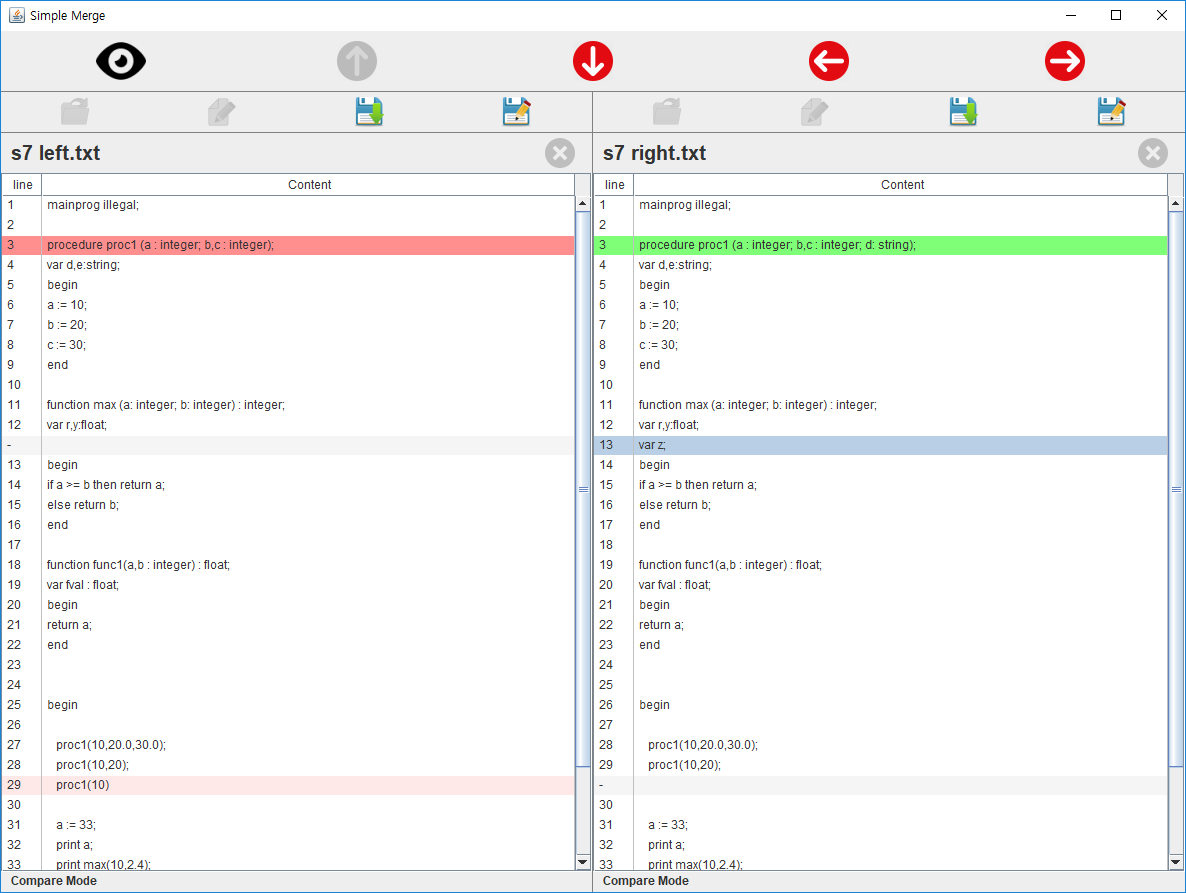
**S5. Result :** 여러 개의 빈 줄 블록이 있음이 나타난다. 또한 왼쪽 파일과 오른쪽 파일에서 다르게 나타나는 부분도 같이 블록에 포함되어 있다.



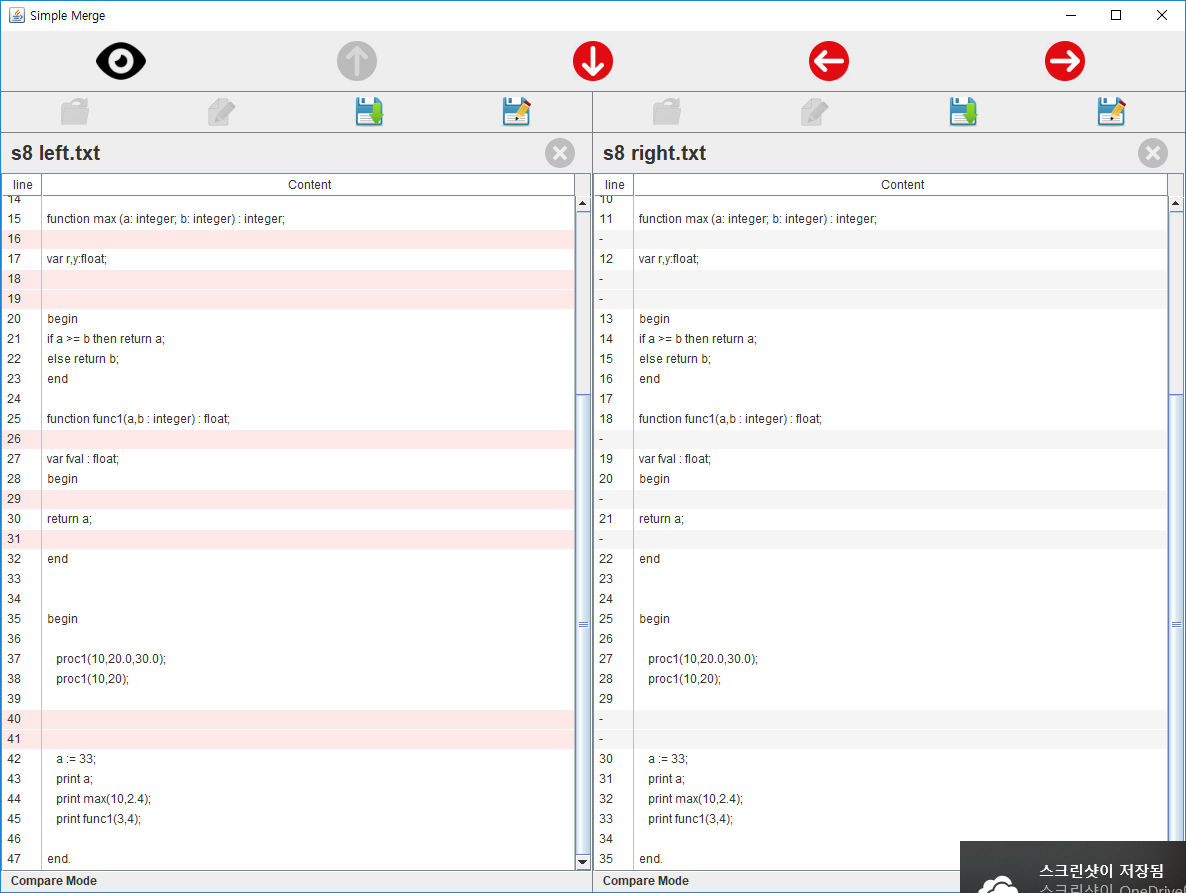
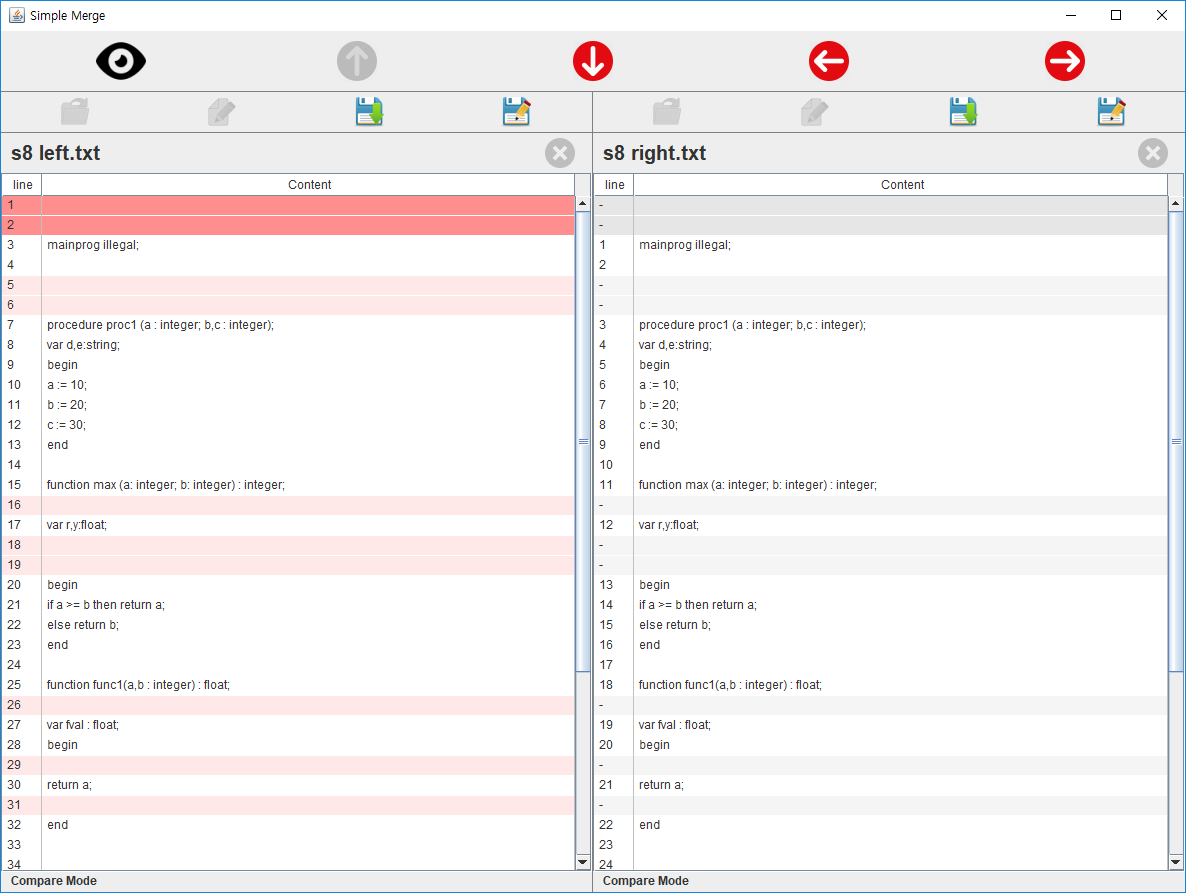
**S6. Result :** 두 파일이 동일함이 나타난다.



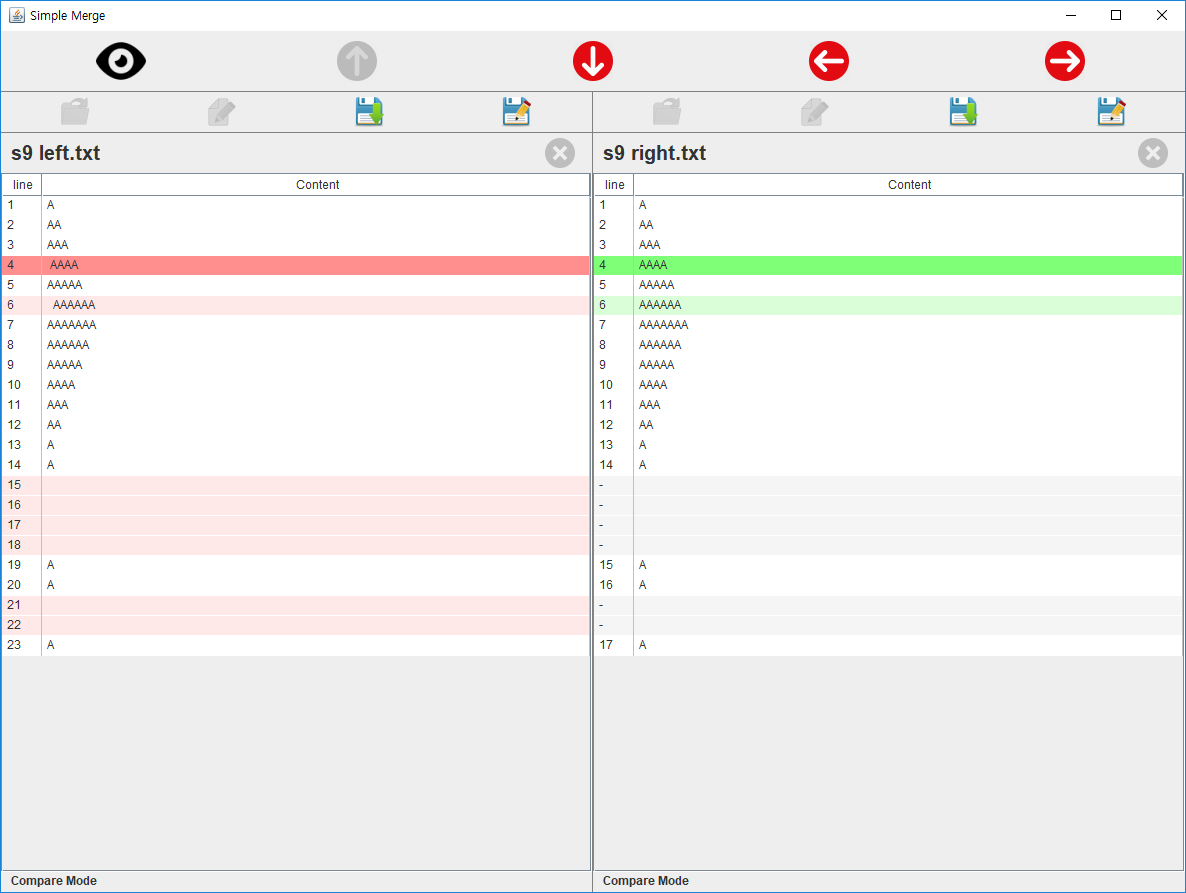
**S7. Result :** 3번째 줄이 다름, 왼쪽 파일에만 빈 줄이 있음, 오른쪽 파일에만 빈 줄이 있음, 37번째 줄이 다름이 나타난다.



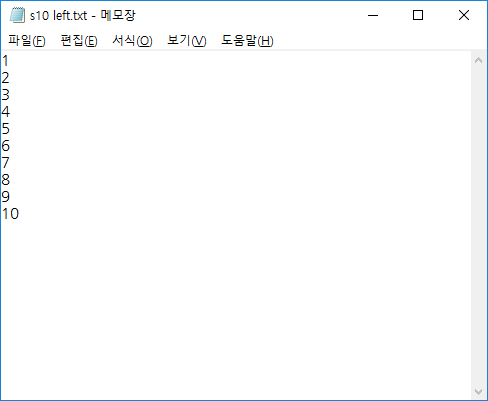
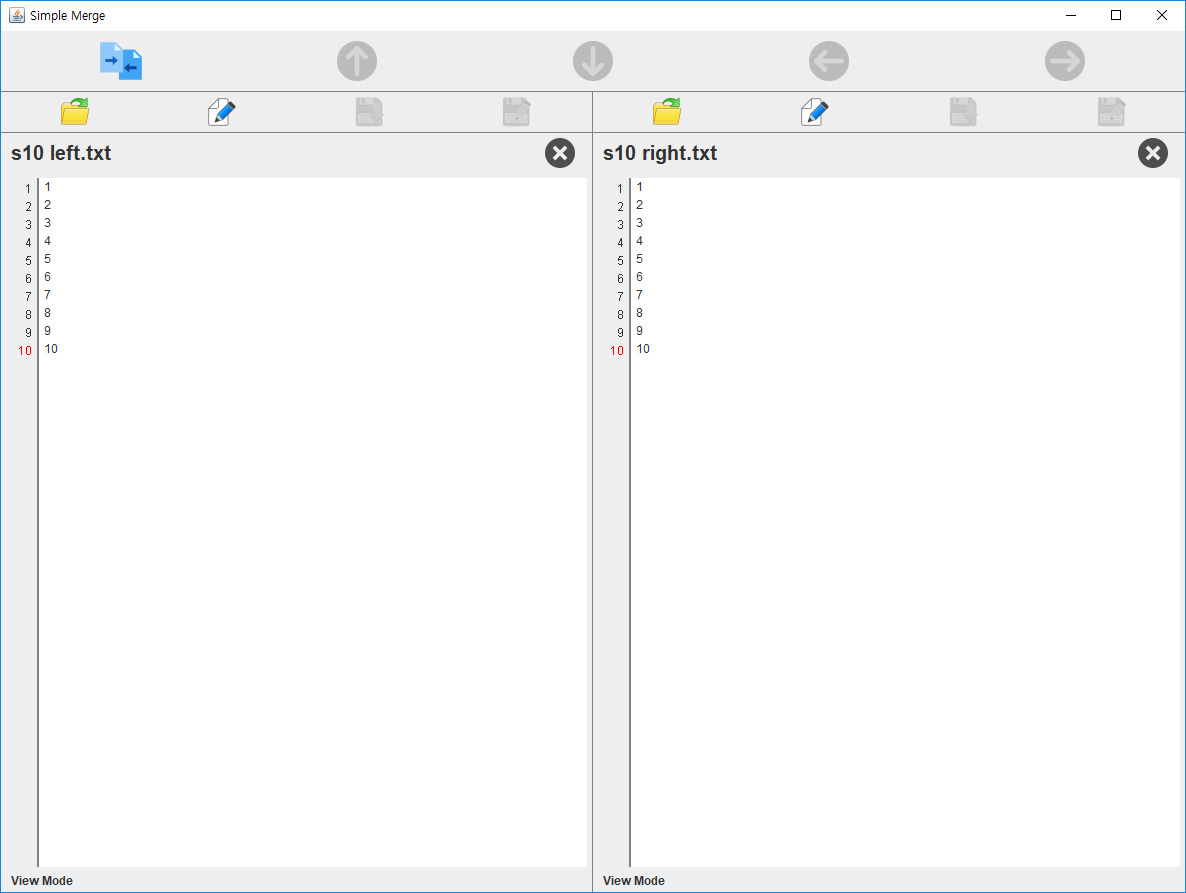
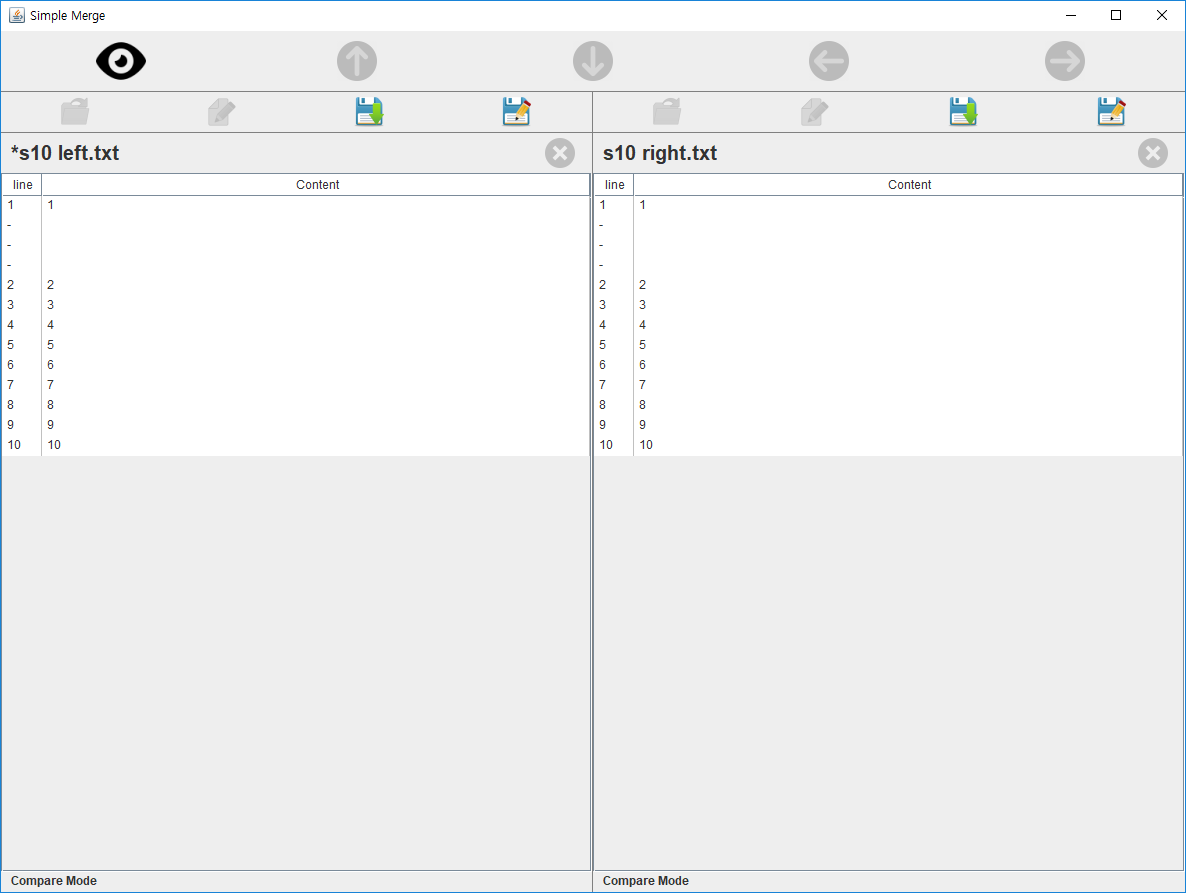
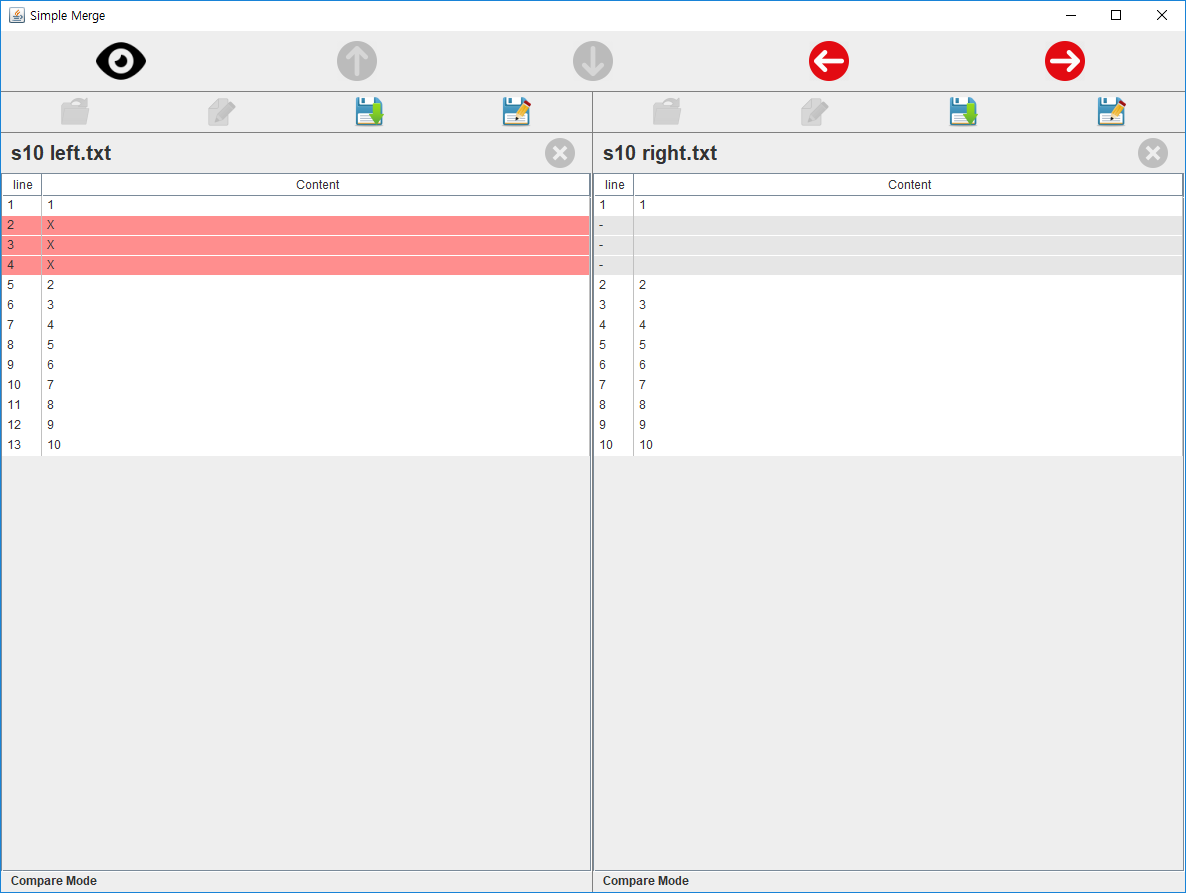
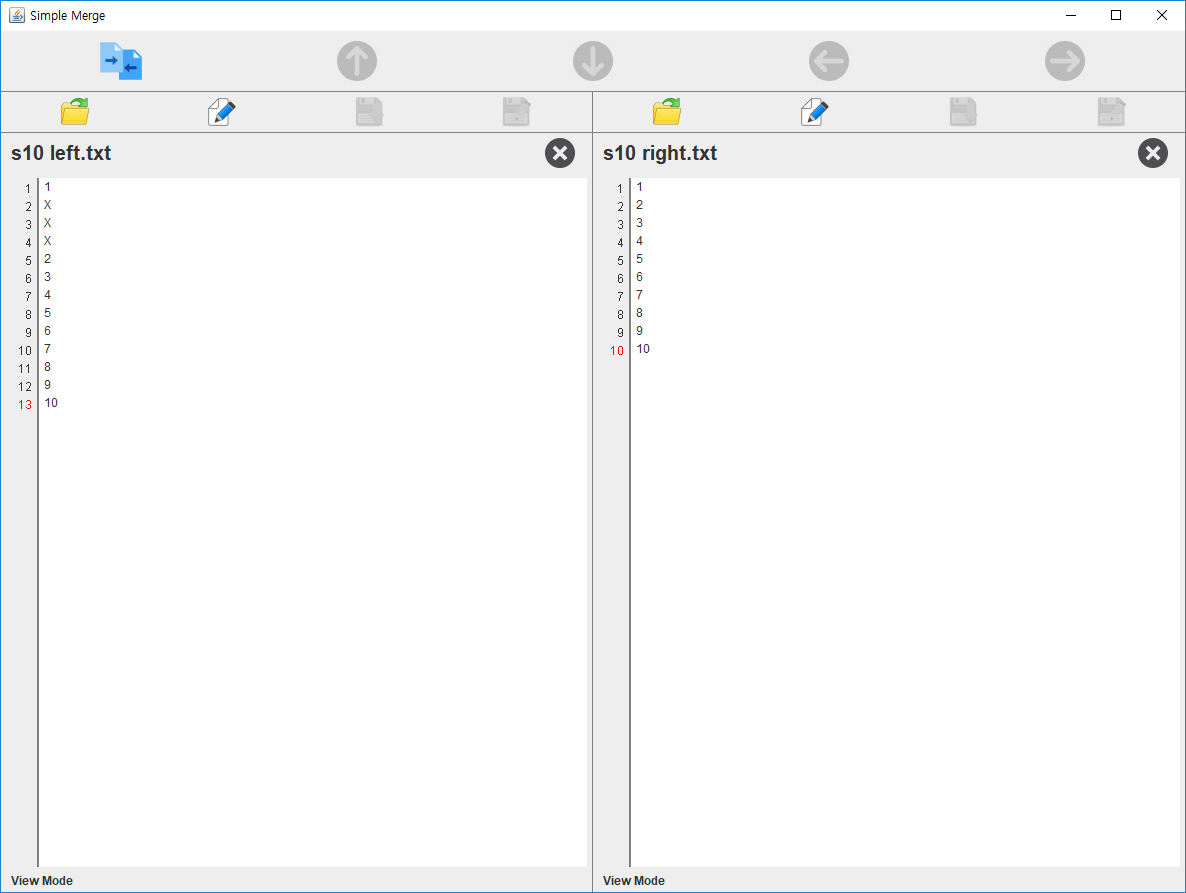
**S8. Result :** 여러 개의 빈 줄 블록이 있음이 나타난다.



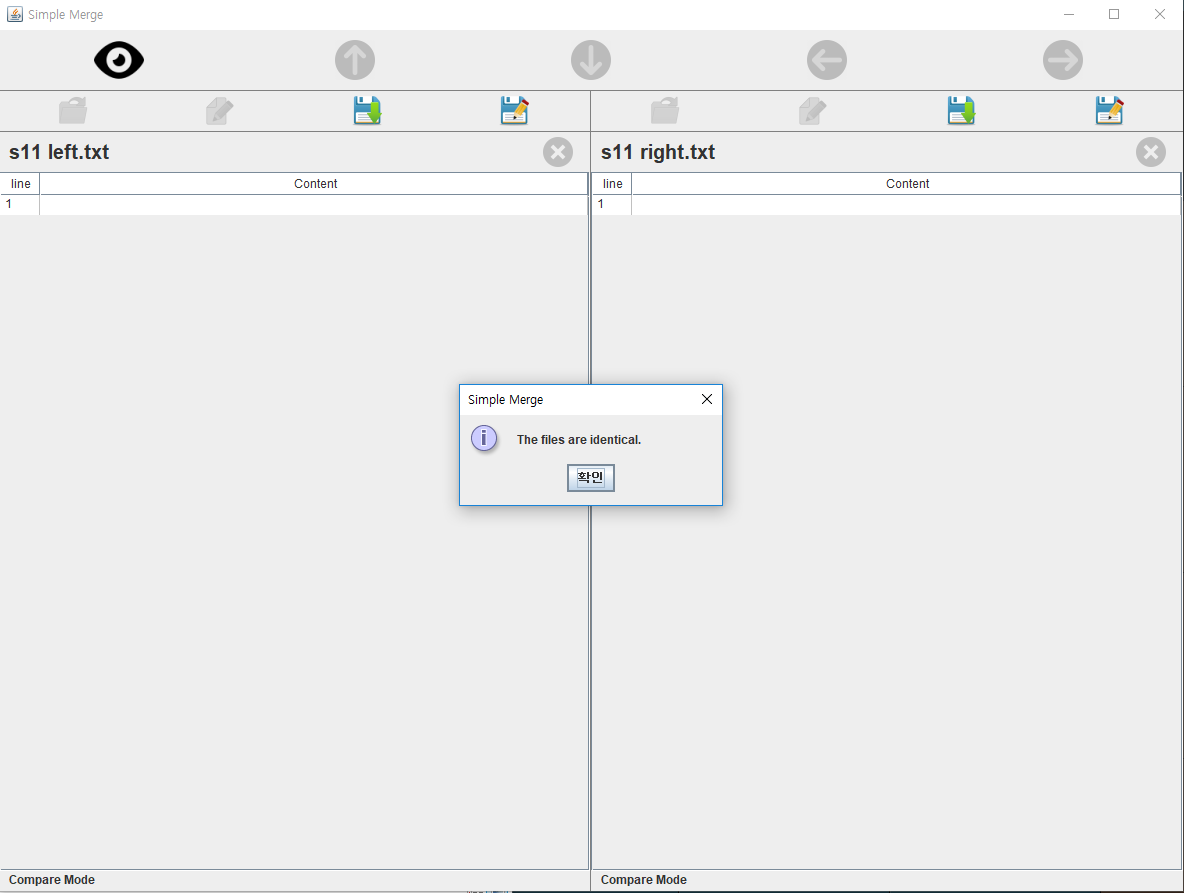
**S9. Result :** 여러 개의 빈 줄 블록이 있음, 4, 6번째 줄이 다름이 나타난다.



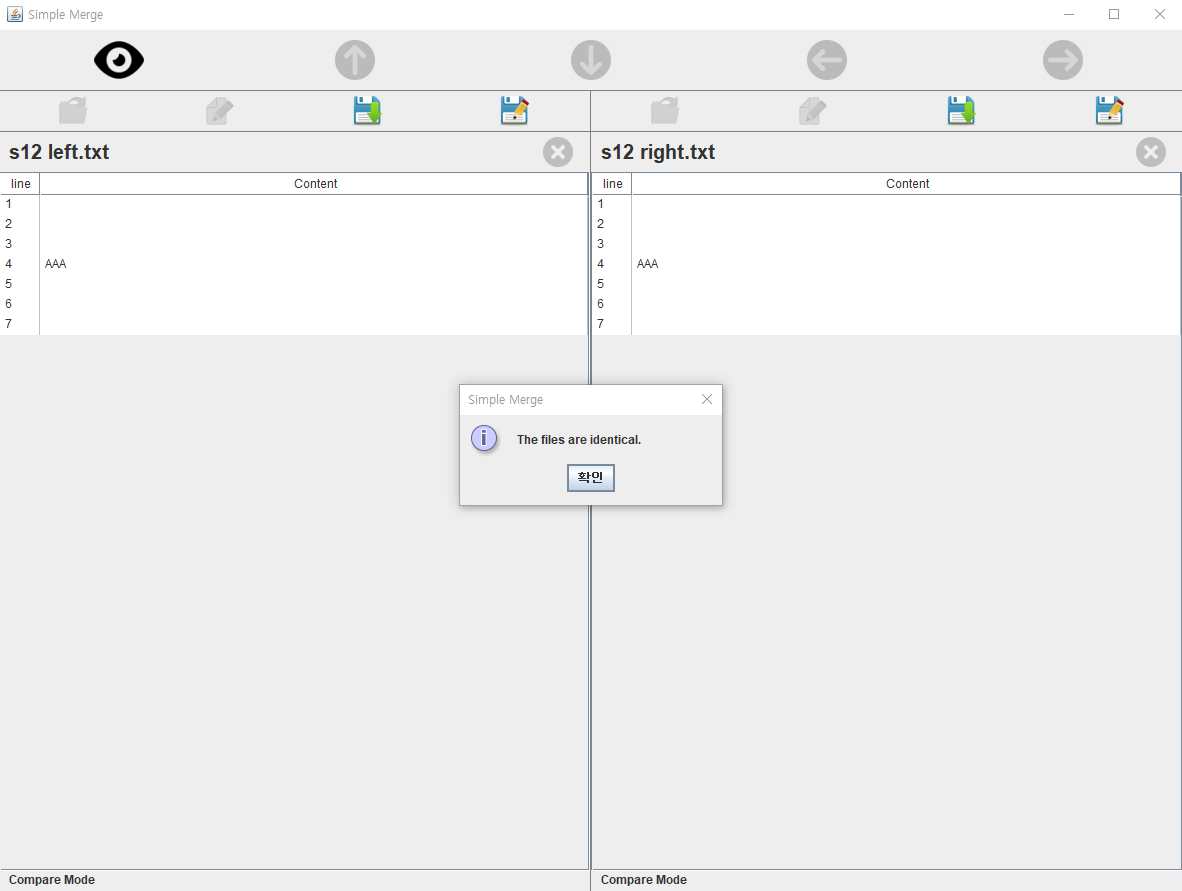
**S10. Result :** Compare 버튼을 누르면 양쪽 파일에서 다른 부분이 색칠되고, Copy to Left 버튼 ( <- ) 을 누르면 3번째 사진과 같이 파일 내용이 동일해지는 것을 볼 수 있다. Save 버튼을 누르고 Compare 모드에서 벗어나기 위해 눈동자 모양의 버튼을 클릭하면 View 모드로 전환되며, 두 파일의 내용이 완전히 동일함을 볼 수 있다. 이후 왼쪽 파일을 메모장으로 켜면 파일의 변경 내용이 제대로 Save 되었는지를 볼 수 있다.



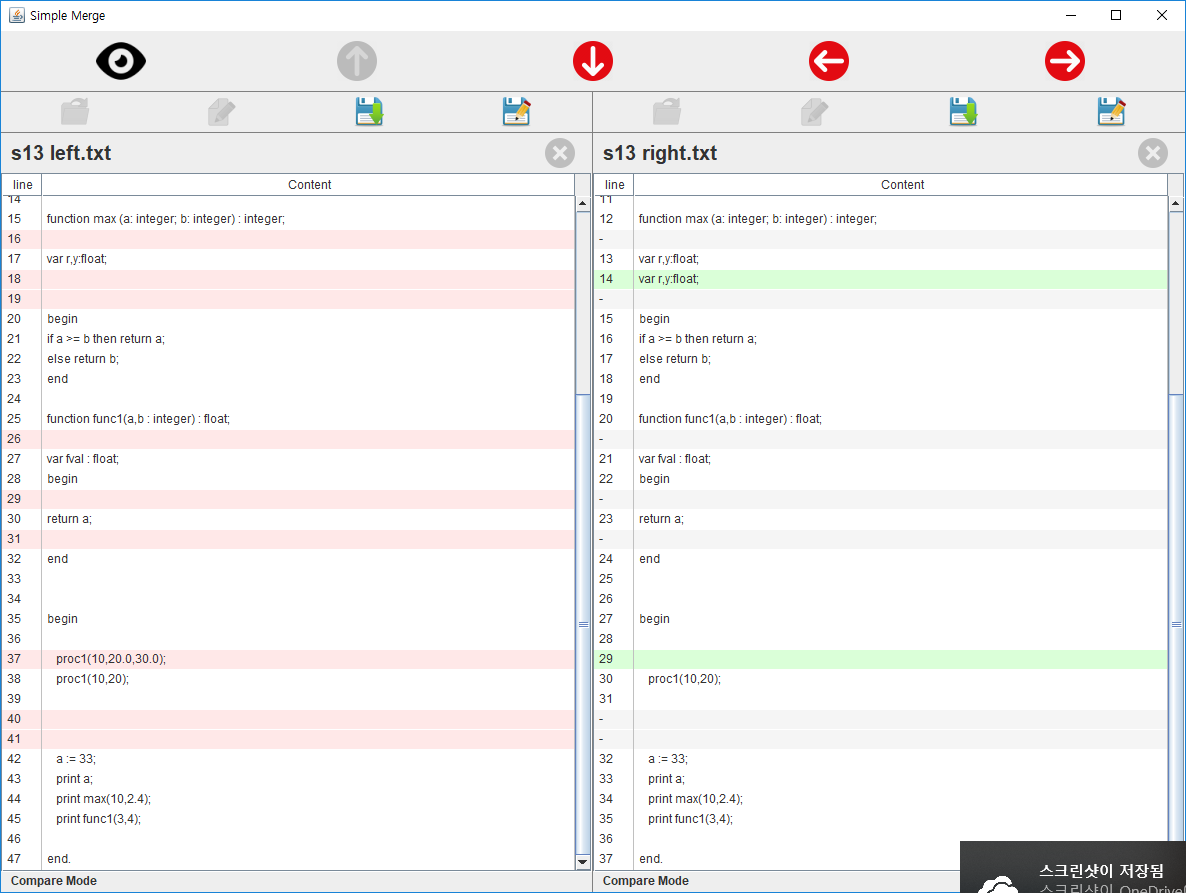
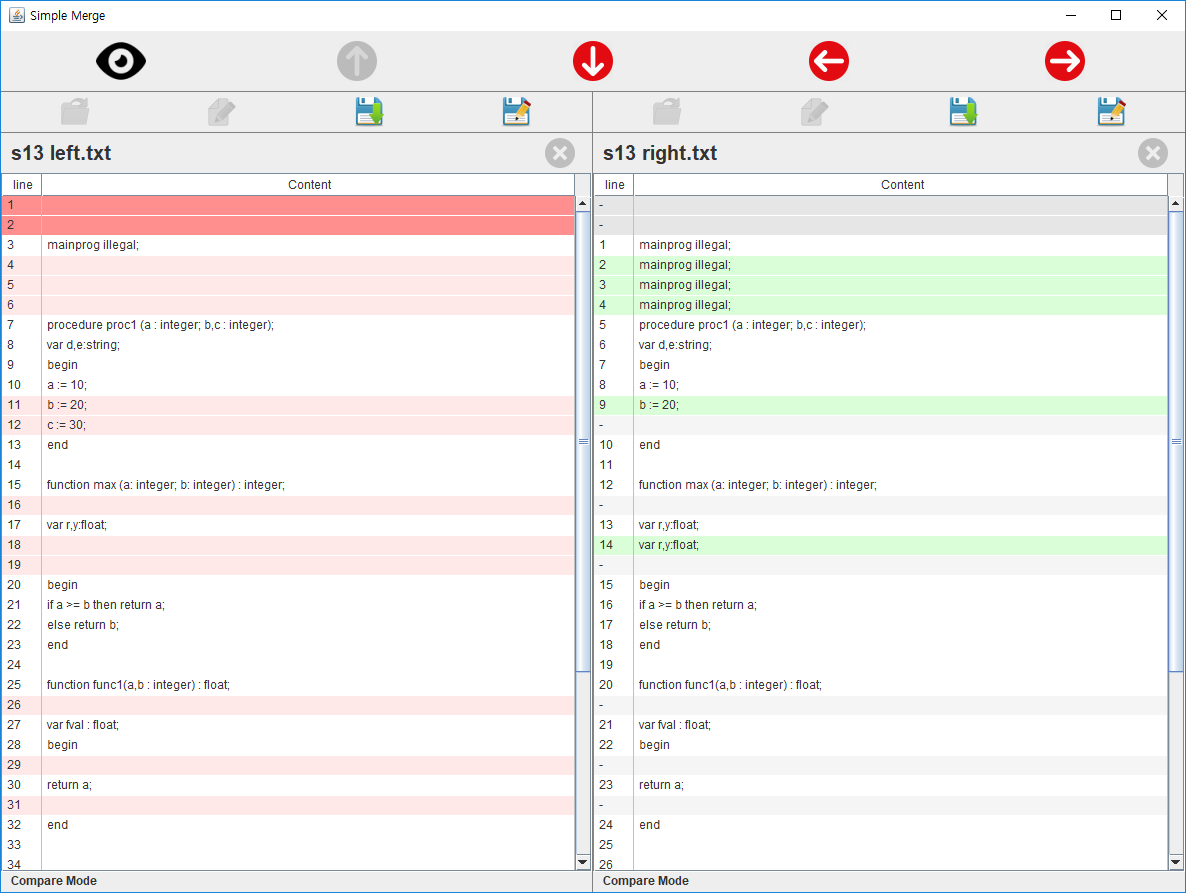
**S11. Result :** 비어 있는 두 파일의 내용이 동일함을 볼 수 있다.



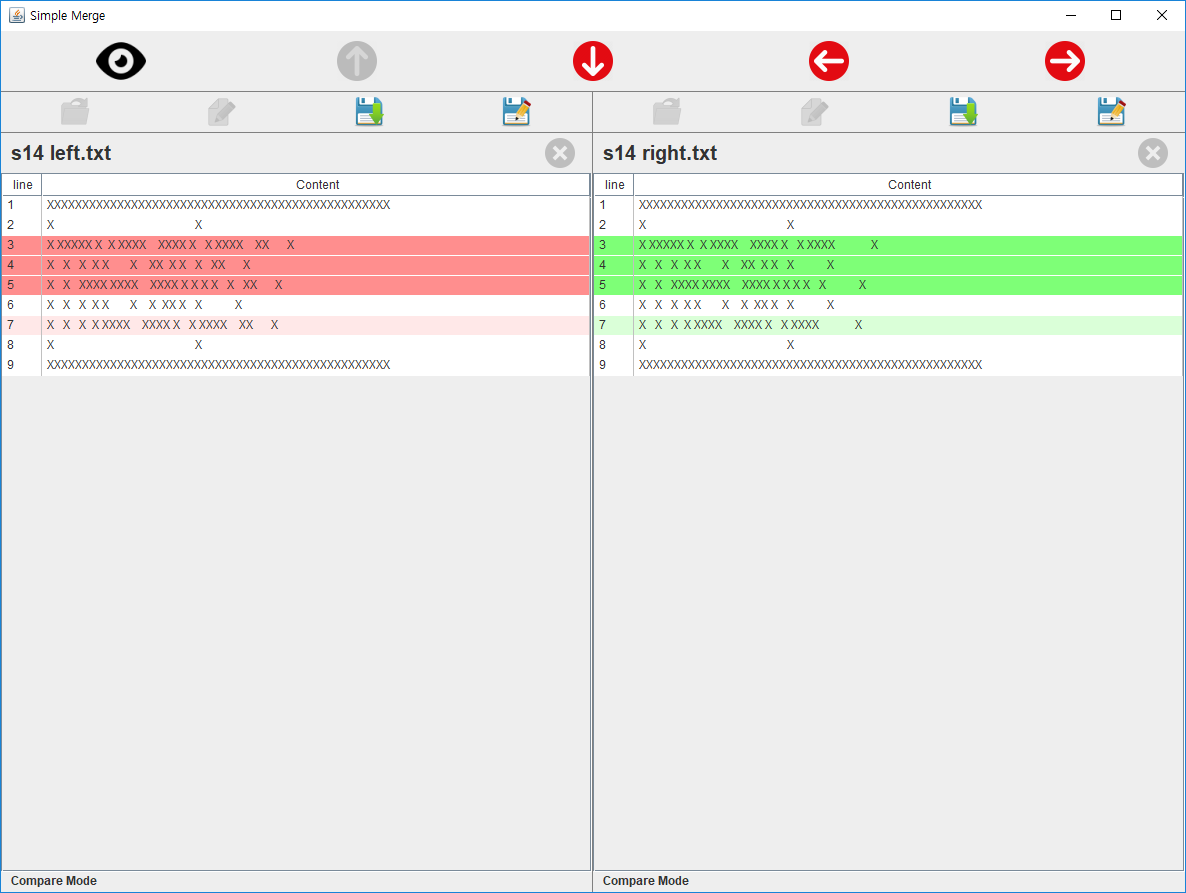
**S12. Result :** 3줄의 빈 줄, 4번째 문장, 다시 3줄의 빈 줄이 나타남을 알 수 있으며, 두 파일의 내용이 동일함을 볼 수 있다.



**S13. Result :** 여러 개의 빈 줄 블록, 다른 내용 블록들이 모두 색칠되어 있는 것을 볼 수 있다.



**S14. Result :** 세 줄 블록의 내용이 다르고, 한 줄 블록의 내용이 달라서 색칠이 되어있음을 알 수 있다.



1. **(If any) functional limitations**