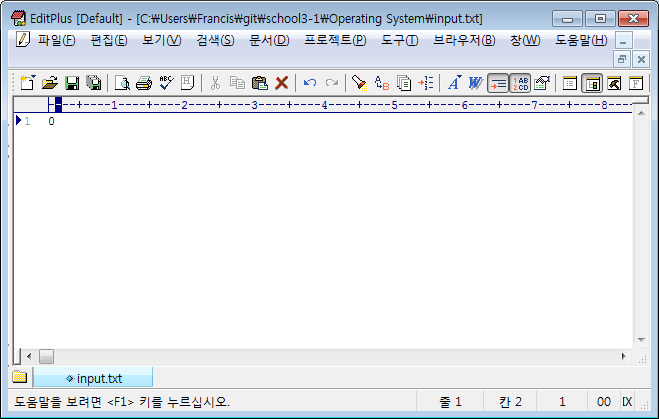
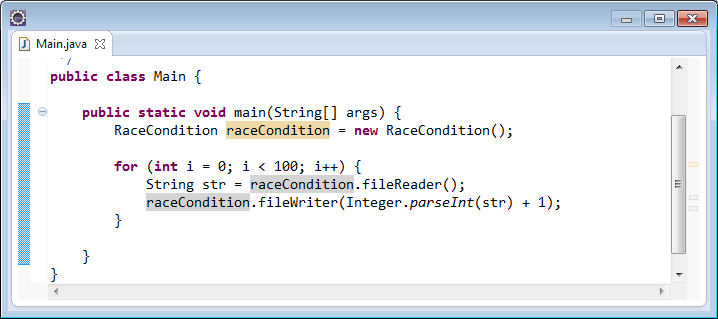
**1. 다음 기능을 하는 프로그램을 만드시오  
1) 데이타 화일의 마지막 숫자를 읽어서 1을 더해서 그 화일의 끝에 추가해서 기록**

**(데이타 화일은 초기에 0만 써있다.)**

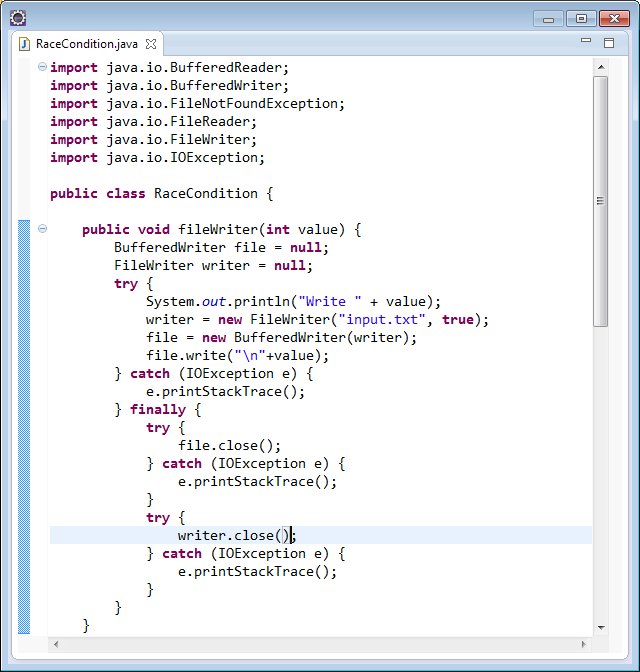
- input.txt



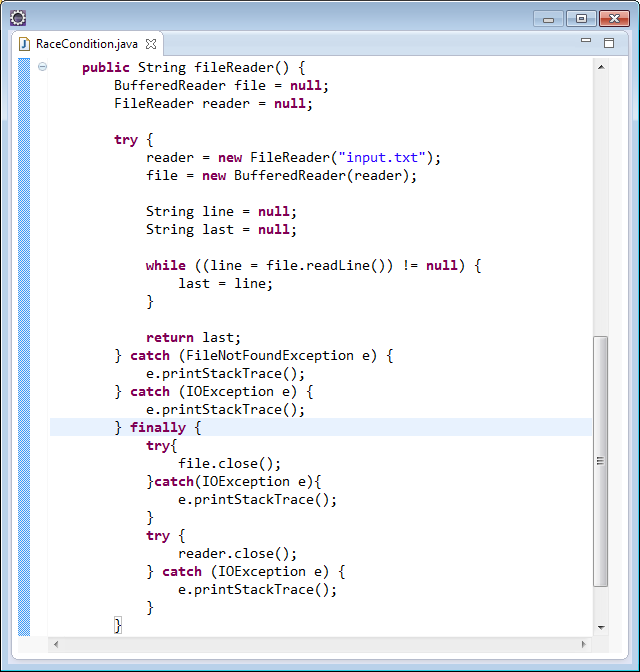
- Main.java ( Main Method )



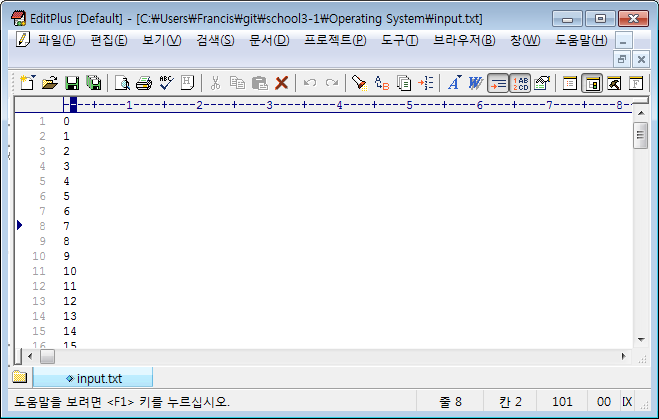
**-** RaceCondition.java ( fileWriter Method - int type Parameter )



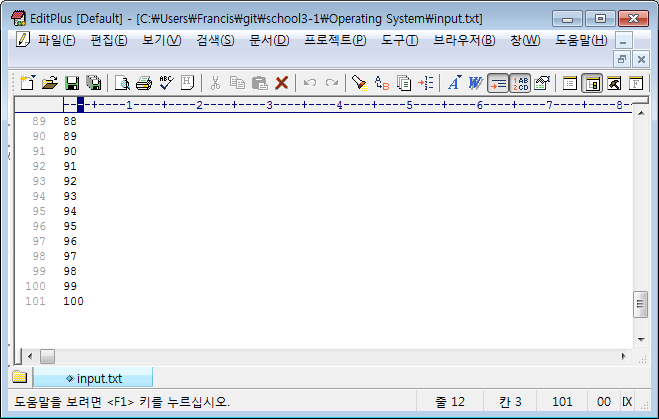
- RaceConition.java ( fileReader Method - String return type)



**2)위 함수를 100번 호출**- input.txt ( 결과 )



**…**



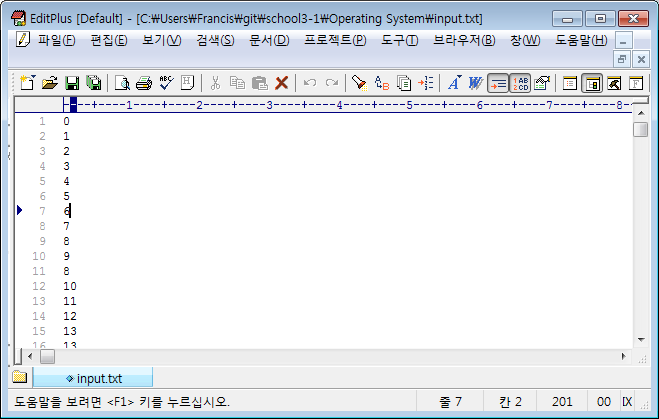
**3)이 프로그램을 실행하면 0 부터 100까지 101개의 숫자가 기록될 것이다.**

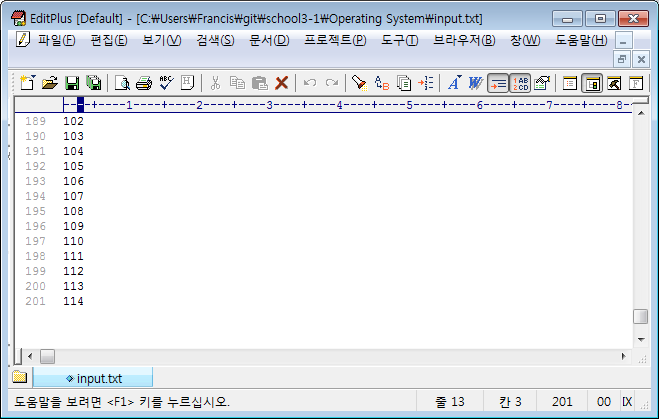
**2. 이 프로그램을 두개 이상 동시에 실행하여 결과를 살펴보시오**

**1) n개를 동시에 실행했을 때 화일의 마지막 숫자가 100\*n인가?**

**예: 2개를 동시에 실행했을 때 200**

input.txt





**2) 결과가 100\*n이 아니라면 그 원인은 무엇인지 설명하시오.**

2개의 프로그램을 실행한다고 가정하고, 두 개의 프로그램을 각각 Program A, Program B 라 가정한다. 이 프로그램을 실행한 결과는 [ 100 \* n ] 이 생성되지 않는다. 하지만 결과물을 보면 201줄로 처음에 존재하였던 0을 제외하면 200번 실행되었음을 알 수 있다. 이것으로 보아 두 프로그램이 정상적으로 실행된 것으로 파악할 수 있고 프로그램이 번갈아 실행되면서 이와 같은 문제가 발생되었다고 유추할 수 있다. 예를 들어 Program A가 먼저 실행되었을 때 [ fileReader ]라는 메소드를 이용해서 [ input.txt ] 파일을 읽은 상황에서 Program B 가 실행되어 동일한 메소드를 이용하여 파일을 읽는 상황이 발생된다. 이렇게 된 경우 값이 증가되지 않고 동일한 값을 취하기 때문에 값이 증가가 이루어지지 않아 200이 되지 않는다. 또한 이렇게 서로 번갈아 가며 reader와 writer를 한다면 [ 1 1 2 2 3 3 ] 과 같이 증가하여 마지막에 100이 떠야 하겠지만, 실행 되는 과정 중에 I/O 에 대한 시간차나 그 외의 상황이 발생하기 때문에 순차적으로 실행 되는 것이 아닌 reader와 writer가 무작위로 실행되는 상황이 발생되어 100 \* n 이 생성되지 않는 상황이 발생된다고 생각할 수 있다.

결론적으로 이러한 상황은 지금까지 배운 사항에 근거하여, I/O 에서 결과완료에 대한 interrupt발생, 대부분 많이 쓰이는 Ready Pool에서의 Priority Queue, Timer로 인한 Interrupt와 같은 것들에 의하여 결과가 위와 같이 뜬다고 생각한다. 이런 상황을 해결하기 위한 방법은 시스템 실행을 Atomic 하게 하기 위해 Lock을 써서 해당 프로세스가 완전히 독립적으로 실행되도록 해야 하며, 이렇게 하면 100 \* n 이 출력 결과물이 출력 될 것이라고 생각된다.