**Checker Readme**

**0. 기본 사용법**

**1. 실행 오류**

**2. 고급 설정**

**3. 테스트케이스 제작**

**0. 기본 사용법**

**.java파일을 YOUR\_FILES에 넣습니다.** (**제출해야 하는 파일만** 넣으면 되며, 헬퍼 코드나 테스트 코드는 같이 안 넣어도 괜찮습니다)

data파일이 없다면 data폴더 안에 넣습니다. (컴퓨터 프로그래밍 18-1의 2-1, 2-2, 3-1, 3-2번 과제는 기본 제공됩니다)

Checker(Windows).bat이나 Checker(Mac).sh를 실행합니다.

X-X형식으로 체크할 과제를 입력합니다. (예- Assignment2 의 Problem1이라면: 2-1)

**1. 실행 오류**

**1) 바로 꺼지는 경우**

**“체커 작동 안할시(Windows).bat”를 실행하면 바로 꺼지지 않습니다.**

이때

**‘javac’은/는 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는 배치 파일이 아닙니다** ->

Windows 10: <https://www.youtube.com/watch?v=-NS1SbiKGtY> 참고

다른 OS는 javac not recognized 라고 검색해서 직접 찾기 바랍니다.

**2) 컴파일 실패**

Java가 해당 버전의 파일을 인식할 수 없다고 뜨는 경우->

Java Runtime을 최신 버전으로 업데이트하시기 바랍니다.

**2. 고급 설정**

**1) 선택적 테스트**

**X-X 4**

테스트케이스 4만 실행

**X-X 3-**

테스트케이스 3이후만 실행

**X-X -3**

테스트케이스 1부터 3까지 실행(3포함)

**X-X 2-5**

테스트케이스 2부터 5까지 실행

\*아래부터는 모두 X-X를 입력하기 전에 입력하는 사항입니다. 대/소문자는 자유입니다.

**2) 기타 옵션**

**skipCleanup**: 테스트 이후 생성된 파일이 삭제되지 않습니다. (파일 확인에 유용)

파일 출력은 YOUR\_FILES에 그대로 남습니다.

System.out의 내용은 data/X-X/테스트케이스번호/sysOut.txt 에 있습니다.

마지막으로 실행한 테스트의 내용만 저장하므로 선택적 실행과 함께 하는 것을 추천합니다.

**enableCleanup**: skipCleanup을 취소합니다(프로그램을 종료해도 취소됩니다).

**Exit**: 프로그램을 종료합니다.

**Credits**: 제작자를 표시합니다.

**Makemode**: 해당 .java파일이 옳다고 가정하고 테스트케이스의 output를 제작하는 제작 모드로 전환합니다.(자세한 것은 후술)

**Disable Makemode**: 일반 모드로 전환합니다.

**setTimeLimit**: 시간제한을 바꿉니다. 시간은 Double형식으로 입력합니다.

**eraseOutput**: 테스트케이스의 성공 여부를 표시하지 않습니다.

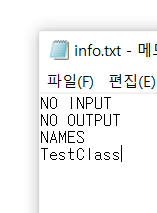
**3. 테스트케이스 제작**

\*본 체커는 임의의 문제에 대한 체커로 사용될 수 있습니다.

X-X 형식의 폴더를 만듭니다. (예: data\1000-1)

안에는 info.txt를 하나 넣습니다. (예: data\info.txt)

1) 파일 입출력이 없는 경우



**NAMES 아래에 테스트대상(.java파일)의 이름을 적습니다.** (.java는 빼고)

테스트케이스 번호를 이름으로 하는 폴더를 만듭니다. (예: data\1000-1\1)

**입력**

해당 테스트케이스의 입력을 **sysIn.txt**로 제공합니다.

**출력**

해당 테스트케이스의 정답을 **ansOut.txt**로 제공하거나,

입력과 정답인 .java파일을 제공한 후, 체커에서 makeMode에 진입 후 X-X라고 입력합니다.

(그러면 ansOut.txt 가 자동 생성됩니다)

실제 다른 data파일을 보면서 따라하면 쉽습니다(0-0 참조)

\*시간제한(선택사항)

맨 아랫줄에

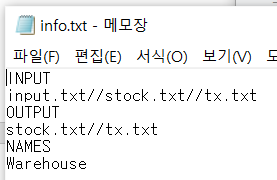
TIME

0.05

과 같이 입력합니다. 시간 단위는 sec이며 Double형식으로 입력합니다.

(입력하지 않을 시 기본 시간제한인 3초가 적용됩니다)

2) 파일 입출력이 있는 경우



INPUT 아래에 파일 인풋을 //로 구분하여 적습니다. (없으면 NO INPUT)

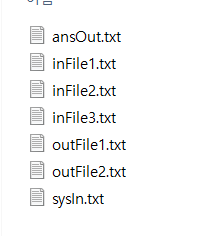
OUTPUT도 마찬가지입니다. (없으면 NO OUTPUT)

같은 파일이 INPUT이자 OUTPUT이어도 괜찮습니다.

INPUT아래에 적은 파일들을 이름을 inFile1.txt, inFile2.txt, inFile3.txt …로 바꿔 넣습니다. (순서대로)

OUTPUT아래에 적은 파일들을 이름을 outFile1.txt, outFile2.txt …로 바꿔 넣습니다. (순서대로)

sysIn과 ansOut도 넣습니다.



예) 위의 예시의 경우 inFile1 = input.txt, inFile2 = stock.txt, inFile3 = tx.txt

outFile1 = stock.txt, outFile2 = tx.txt 입니다. (순서 유지)