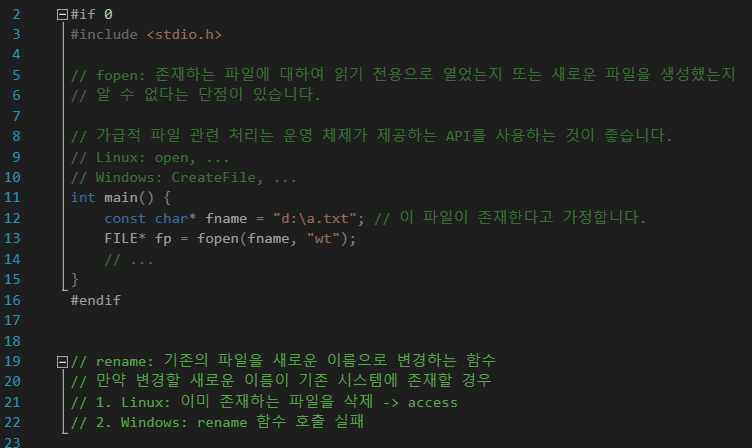
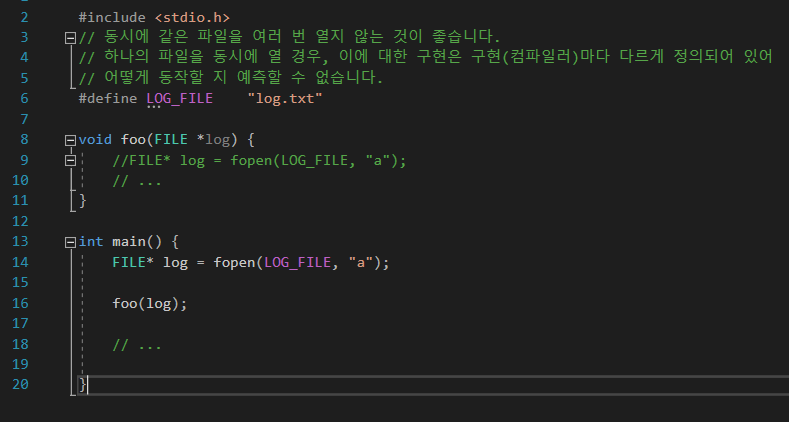


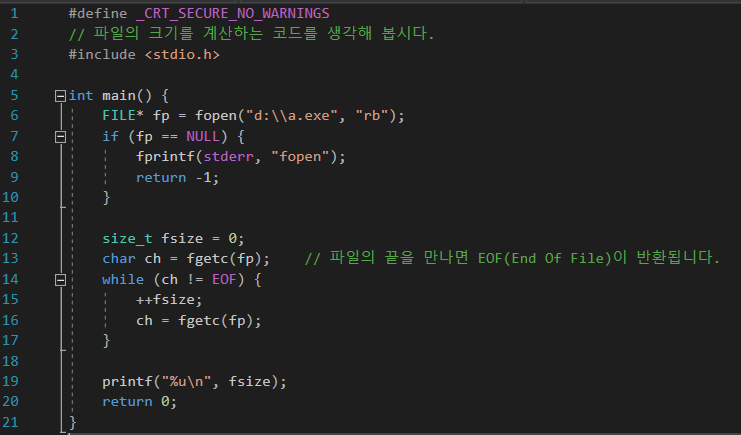
포맷 문자열 사용시, 정확한 문자열을 사용해야 한다. 잘못된 포맷 문자열 사용시 미정의 동작 처리된다.



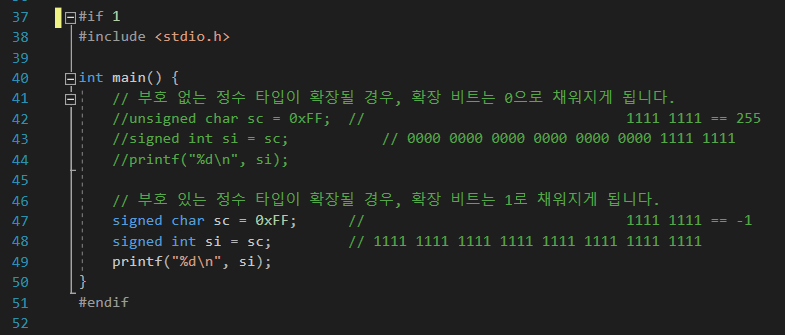
fopen 함수는 존재하는 파일에 대하여 읽기 전용으로 열었는지 또는 새로운 파일을 생성했는지 알수 없다는 단점이 있다.



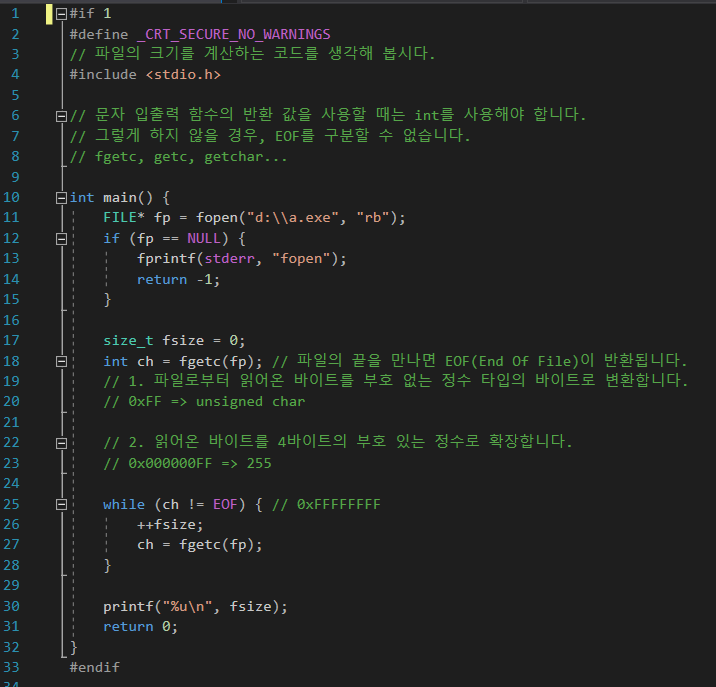
동시에 같은 파일을 여러 번 열지 않는 것이 좋다. 하나의 파일을 동시에 열 경우, 이에대한 구현은 구현(컴파일러)마다 다르게 정의되어 있어 어떻게 동작할지 예측할 수 없다.

일반 txt 파일은 char로 하나씩 읽어오면 정상적인 값이 출력된다.

그러나 a.exe같은 실행파일은 char로 하면 정상적인 값이 안나오며, 반환타입을 int로 해야 정상적인 값이 출력된다(위 코드는 오류)

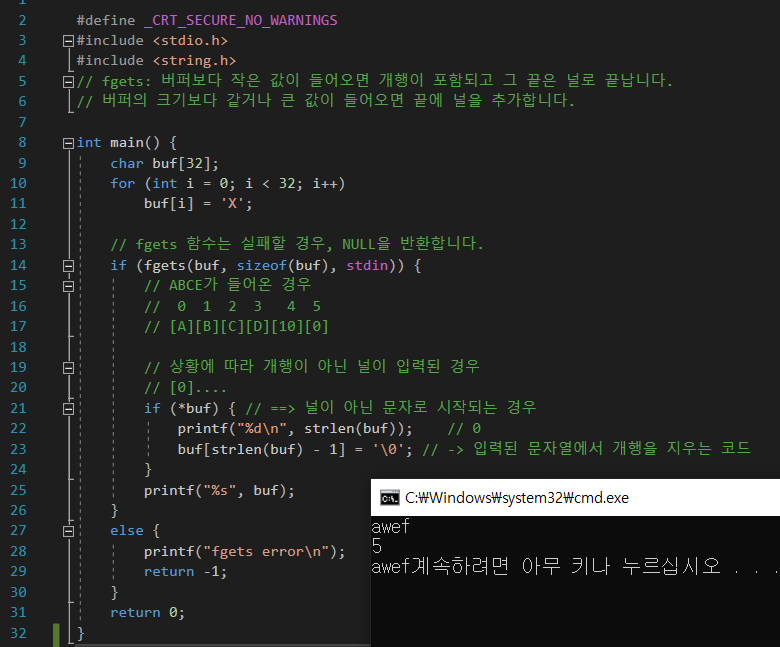
signed char 로는 FF와 EOF 를 둘다 -1로 취급하므로 정상적인 값이 나오지 않는다.

🡺결론! : 바이트 타입으로 읽어올때는 char형이아니라 int형으로!! (char 는 oxFF와 EOF를 구분할수가 없다..)



문자 입출력 함수의 반환값을 사용할때는 int를 사용해야한다.

그렇게 하지 않을경우 EOF를 구분할수 없다.

fgets의 주의 사항(fgets 함수는 실패할 경우 null을 반환한다)

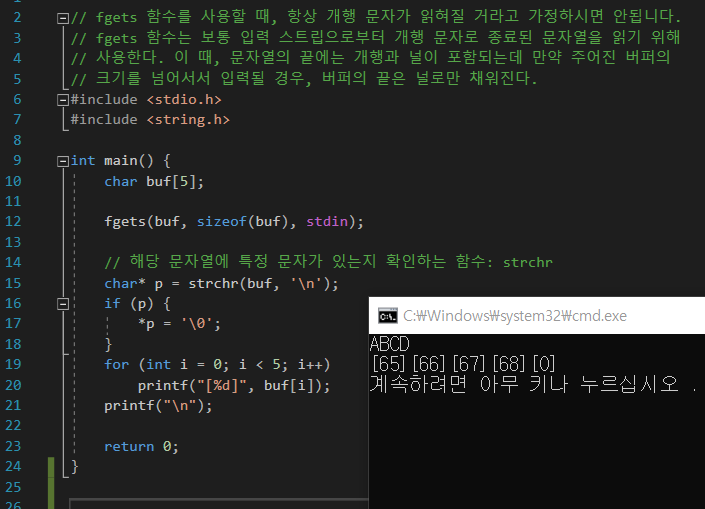
fgets – 버퍼보다 작은 값이 들어오면 개행이 포홤되고 그 끝은 널로 끝난다.

버퍼의 크기와 같거나 큰 값이 들어오면 끝에 널을 추가한다. (ex) ABCD 65 66 67 0)

fgets – 스트림에서 문자열을 받아온다

fputs – 스트림에 문자열을 쓴다.

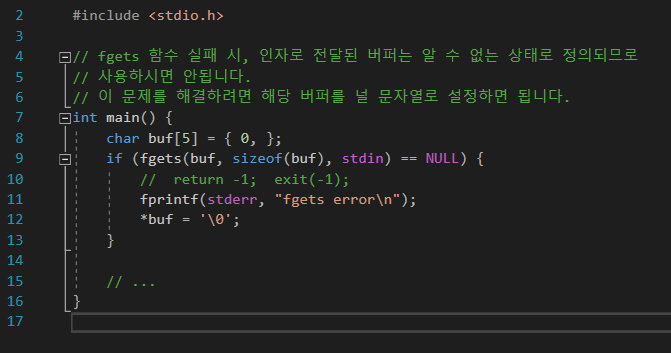
Day0526

해당 문자열에 특정 문자가 있는지확인하는 함수: strchr

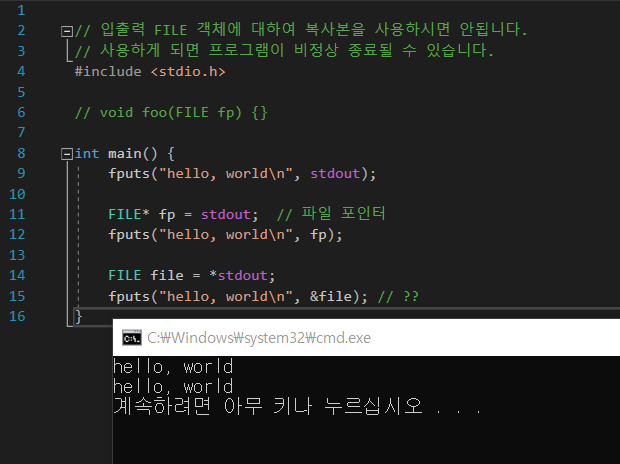
위 예제에서는 \n을 제일 처음 발견하는 순간 char \* 로 리턴함

->fgets함수를 사용할 때 항상 개행문자가 읽혀질 거라고 가정하면 안된다.

fgets함수는 보통 입력 스트림으로부터 개행문자로 종료된 문자열을 읽기 위해 사용한다. 이때 문자열의 끝에는 개행과 널이 포함되는데 만약 주어진 버퍼의 크기를 넘어서서 입력될 경우, 버터의 끝은 널로만 채워진다.

fgets함수가 실패시, 인자로 전달된 버퍼는 알수 없는 상태로 정의되므로 사용하면 안된다.

이 문제를 해결하기 위해서는 해당 버퍼를 널 문자열로 설정하면 된다.



입출력 FILE 객체에 대하여 복사본을 사용하면 안된다. 사용하게 되면 프로그램이 비정사 종료될수 있다.