**[ 실습 1 ]**

**파일명 : fileioLab1.py**

**다음 내용을 생성해서 c:/iotest/today.txt 라는 파일에 저장한다.**

**c:/iotest 디렉토리의 존재여부를 채크하고 존재하지 않으면 새로이 생성한다.**

**출력 내용은 다음과 같습니다. (XXX는 본인의 별명(^^))**

**오늘은 2020년 07월 23일입니다.**

**오늘은 목요일입니다.**

**XXX는 X요일에 태어났습니다.**

**파일에 위의 내용을 저장한 다음에 화면에는“저장이 완료되었습니다.”를 출력한다.**

**[ 실습 2 ]**

**파일명 : fileioLab2.py**

**제공된 sample.txt 파일을 읽고 sample\_yyyy\_mm\_dd.txt 파일에 그대로 출력하는 프로그램을**

**구현한다. 이 파일은 append 모드로 오픈하여 여러 번 테스트하면 sample.txt 파일의 내용이**

**sample\_yyyy\_mm\_dd.txt 파일에 계속 추가되게 한다.**

**위의 내용을 파일에 저장하는 기능이 정상적으로 수행되면 화면에 “저장이 완료되었습니다.”**

**FileNotFoundError 발생시 FileNotFoundError 에 대한 메시지 정보를 출력하고 종료한다.**

**[ 실습 3 ]**

**파일명 : stLab1.py**

**프로그램 수행 중간에 년도와 월을 입력받아 해당년과 월의 달력을 출력한다.**

**년도와 월은 숫자로 입력받아야 하며 숫자가 입력되지 않아서 발생하는 에러를 대비해야**

**한다. ValueError 처리를 통해서 숫자가 입력되지 않은 경우에는 메시지를 내보내고**

**다시 입력받는다. 잘 할때까지……**

**년도와 월이 제대로 입력되면 해당 년월의 달력을 출력한다.**

**[ 실습 4 ]**

**파일명 : fileioLab3.py**

**제공된 yesterday.txt 파일을 읽고 이 문서에 "Yesterday" 가 몇 개 들어 있는지 개수를 세어서 출력한다. (이 때 대소문자는 구분하지 않는다.)**

**FileNotFoundError 발생시 파일을 읽을 수 없어요 를 출력하고 종료한다.**

**에러가 발생하지 않으면 Number of a Word 'Yesterday' X 를 출력하고 종료한다.**

**에러 발생 여부와 관계없이 수행완료!! 를 출력하고 종료한다.**

**(try-except-else-finally 를 모두 사용해서 해결한다.)**