

2020년 2학기 C실습 1분반 최종 테스트

제출마감 : 2020년 12월 20일(일) 밤 12시

제출형태 : 동영상 파일 및 온라인 설문 응답

제출링크 : <https://forms.gle/xLRVU1idhnpvJ5sh7>

제공소스 : <https://github.com/kkim-hgu/2020FallCLabs/tree/main/FinalTest>

참고 실행동영상 : <https://hducc.handong.edu/em/5fd1b171fb236>

1. 동영상 제작 방법

- 제공소스 URL에서 2개의 파일을 다운받아 자신의 코드 제작 에디터에 붙여넣는다.
- 줌의 자신의 계정 <설정>에서 <클라우드 기록> <고급 클라우드 기록 설정> <자동 기록> 부분의 항목을 모두 체크한다.
- 녹화 전에 문제를 해결하기 위한 생각을 충분히 하면서 **미리 제공된 소스의 적절한 부분에 생각을 메모로 적어둔다.**
- 줌 회의를 하나 열고 화면 공유를 클릭하여 <화면> 전체를 공유한다.
- 마이크와 카메라를 켜고 "문제풀이를 시작하겠습니다" 멘트로 시작하면서 문제풀이를 시작한다.
- 문제풀이가 종료되어 테스트 화면을 보여주는 과정을 마치고 나면 "마쳤습니다" 멘트를 하고, 줌 회의를 종료한다.
- 몇시간 후 줌 사이트의 자신의 계정 로그인 후, <기록> 메뉴의 동영상을 다운받아 소요 시간(시간/분/초 단위)을 확인하고, 온라인 설문응답을 통해 파일을 업로드 한다.
- 업로드할 동영상 : 데스크탑화면과 자신의 얼굴이 동시에 나오는 동영상

2. 제작할 프로그램 설명

- 제목 : 서재 관리 프로그램
- 프로그램 설명 : 나의 서재에 보관중인 도서에 대해 보관 및 대여/반납을 관리하는 프로그램을 제작하라.
- 도서정보 구조체
 - struct st_book{
int category; // 분류코드 (0:전공, 1:교양, 2:취미)
char title[100]; // 도서명
char pub[100]; // 출판사
int year; // 출간년도
int status; // 상태코드 (0:보관, 1:대여, 2:분실)
char borrower[100]; // 대여자명
}
- 전역변수
 - char CATEGORY[3][20] = {"전공", "교양", "취미"};
 - char STATUS[3][20] = {"보관", "대여", "분실"};
- main() 에서 사용하는 주요 데이터
 - struct st_book* mylibrary[100]; // 내 라이브러리 (도서정보 구조체 포인터 배열, 최대 100권)
 -
- 이미 구현된 기능
 - 도서정보 읽어오기 - load_books()
 - "books.txt"로부터 분류코드, 도서명, 출판사, 출간년도 정보를 가져온다.

- 현재 서재 정보 출력하기 - view_library()
 - 현재 등록된 서재 내 모든 도서 정보를 출력한다.
- 구현해야 하는 기능
 - [함수1] 도서 추가하기 - add_book()
 - 분류코드, 도서명, 출판사, 출간년도 정보를 입력받아 서재에 등록한다.
 - [함수2] 도서 분실 처리하기 - lost_book()
 - 도서 번호를 입력받아 해당 도서를 분실로 처리한다.
 - [함수3] 도서 검색 - find_books()
 - 검색 방법은 다음 5가지 중 최소 3가지이다.
 - 도서 이름 일부(2글자 이상)를 입력받아 도서 정보를 출력한다.
 - 출판사 이름 일부(2글자 이상)를 입력받아 도서 정보를 출력한다.
 - 출간년도를 입력받아 도서 정보를 출력한다.
 - 도서상태(숫자)를 입력받아 해당 도서들의 정보를 출력한다.
 - 카테고리(숫자)를 입력받아 해당 도서들의 정보를 출력한다.
 - [함수4] 도서정보 수정하기 - update_book()
 - 도서번호를 입력받아 해당 도서의 정보를 수정한다.
 - [함수5] 서재 정보 보고서 만들기 - save_report()
 - 현재의 서재 정보를 library.txt 파일에 저장한다.

3. 함수별 구현 방법 설명

- [함수1] 도서 추가하기 - add_book()
 - 분류코드, 도서명, 출판사, 출간년도 정보를 입력받아 서재에 등록한다.
 - 함수원형
 - void add_book(struct st_book* list[], int* n);
 - 파라미터 : 도서정보 구조체포인터배열, 도서갯수변수 포인터
 - 리턴값 : 없음
 - 필수 구현 내용
 - 추가할 도서정보 구조체의 메모리 할당
 - 입력정보 : 도서명, 출판사, 출간년도, 분류코드
 - 초기 상태는 “보관”으로 처리
 - 도서갯수 포인터를 사용해서 main()의 도서 갯수를 증가시킴
- [함수2] 도서 분실 처리하기 - lost_book()
 - 도서 번호를 입력받아 해당 도서를 분실로 처리한다.
 - 함수원형
 - void lost_book(struct st_book* list[], int n);
 - 파라미터 : 도서정보 구조체포인터배열, 도서갯수
 - 리턴값 : 없음
 - 필수 구현 내용
 - 입력 정보 : 도서번호
 - 해당 번호 도서 상태를 “분실”로 변경
 - 도서 번호 유효성 검사
 - 분실변경 처리를 한번 더 확인 (Yes/No)
- [함수3] 도서 검색 - find_books()
 - 3가지 종류의 검색 방법을 통해 도서를 검색한다.
 - 함수원형
 - void find_books(struct st_book* list[], int n);
 - 파라미터 : 도서정보 구조체포인터배열, 도서갯수

- 리턴값 : 없음
- 도서 검색 방법은 다음 중 **3~5가지를 자유로 선택할 것**
 - 도서 이름 일부(2글자 이상)를 입력받아 도서 정보를 출력한다.
 - 출판사 이름 일부(2글자 이상)를 입력받아 도서 정보를 출력한다.
 - 출간년도(숫자)를 입력받아 도서 정보를 출력한다.
 - 도서상태(숫자)를 입력받아 해당 도서들의 정보를 출력한다.
 - 카테고리(숫자)를 입력받아 해당 도서들의 정보를 출력한다.
- 필수 구현 내용
 - 검색방법 선택
 - 방법에 따라 문자열 또는 숫자로 해당 검색 키워드를 입력받음
 - 검색된 도서 정보 및 검색 갯수 출력
- [함수4] 도서정보 수정하기 - `update_book()`
 - 도서번호를 입력받아 해당 도서의 정보를 수정한다.
 - 함수원형
 - `void update_book(struct st_book* list[], int n);`
 - 파라미터 : 도서정보 구조체포인터배열, 도서갯수
 - 리턴값 : 없음
 - 수정기능 선택
 - 도서 정보 수정 - 도서명/출판사/출간년도/분류코드 를 새로 입력받아 수정함
 - 도서 대여 처리 - 대여자 이름을 입력받아 도서 상태를 "대여"로 변경함
 - 도서 반납 처리 - 도서 상태를 "보관"으로 변경함
 - 필수 구현 내용
 - 수정기능 종류 선택
 - 종류에 따라 적절한 내용을 입력받음
 - 예외 처리 - 대여되지 않은 도서는 반납 불가!, 보관중인 도서가 아니면 대여 불가!
- [함수5] 서재 정보 보고서 만들기 - `save_report()`
 - 현재의 서재 정보를 `library.txt` 파일에 저장한다.
 - 함수원형
 - `void save_report(struct st_book* list[], int n);`
 - 파라미터 : 도서정보 구조체포인터배열, 도서갯수
 - 리턴값 : 없음
 - 필수 구현 내용
 - 도서 전체 갯수
 - 모든 도서 정보 (번호/분류/도서명/출판사/출간년도/상태/대여자)
 - 분류별 도서 갯수
 - 상태별 도서 갯수

4. 주의사항

- 제작 시작부터 테스트 종료까지 동영상 편집 불가함
- 반드시 줌 화면공유는 전체 데스크탑 화면이어야 함. 특정 영역이나 프로그램 화면 공유 안됨!
- `main()` 및 미리 제공된 2개 함수 코드는 수정 불가!
- 화면에 출력하는 문장이나 단어는 실행화면을 참고하되, 임의로 정해도 됨
- 제공된 소스를 다운받아 충분히 검토하고, 생각을 정리하여 소스파일에 메모를 적는다. 단, 코드를 미리 적어두는 것은 불가! 다른 메모장이나 소스파일에서 코드 복붙하는 것은 불가!
- 동영상 녹화 중에 작성하고 있던 같은 창 내의 소스 일부를 다른 부분에 복붙하는 것은 가능함! (예: `load_books()` 함수의 코드 일부를 `add_book()`에 복붙하는 것)