1) (10 pts) DSN (Dynamic Memory Management in C)

Consider the following typedef struct definition that represents a book.

```
//struct representing a book with content
typedef struct {
    char ** sentences; // actual sentences
    int numSentences; // total number of sentences
    char * title; // book title
    char * author ; // book author
} book_t;
```

Complete the following user defined function definition that properly deallocates all memory associated with the heap space of the struct type `book_t`. The parameters of the function contains the reference to heap space of where the **array** of book_t is stored along with the total number of elements as `numBooks`. Note that within each type `book_t`, `sentences` is an array of `numSentences` strings, where each string was dynamically allocated, as were the strings `title` and `author`.

```
void cleanUp(book_t * lib, int numBooks){


    for(int x = 0; x < numBooks; x++)                   // 1 pt
    {
        for(int y = 0; y < lib[x].numSentences; y++)    // 2 pts
            free(lib[x].sentences[y]);                  // 2 pts

        free(lib[x].sentences);                         // 1 pt
        free(lib[x].title);                             // 1 pt
        free(lib[x].author);                            // 1 pt
    }

    free(lib);                                          // 1 pt

     // 1 pt for correctly using . all the time, so just 1 pt
     // off total if -> was used at all.




}
```