Prelab 9

For this prelab, you will implement a mini "library system" for managing book records using structs and efficient searching techniques. Each Book struct will have the following integer and string members:

```
typedef struct {
    int ISBN;    // International Standard Book Number
    int bookID;  // Unique internal book identifier
    char *title; // Dynamically allocated string for the book title
} Book;
```

You will implement functions that build and manage a searchable **BookCatalog** from an array of Book * pointers. The BookCatalog should allow for efficient (O(log N)) retrieval of books by either their ISBN or their internal book ID, and support resource management.

Functions to implement:

- `BookCatalog createSearchableBookCatalog(Book **, int);`
- `Book * findBookByISBN(int, BookCatalog);`
- `Book * findBookByID(int, BookCatalog);`
- `void freeBookCatalog(BookCatalog);`
- `int getBookErrorCode(BookCatalog);`

Requirements:

- The **`createSearchableBookCatalog`** function takes an array of Book * pointers and the number of books, and returns a **`BookCatalog`** that allows for efficient searching. If it fails to allocate memory or initialize correctly, it returns a **`NULL`** BookCatalog.
- The **`findBookByISBN`** and **`findBookByID`** functions should perform lookups in ***O(log N)*** time, meaning you must sort and store auxiliary arrays (e.g., an array of pointers sorted by ISBN and another sorted by bookID) for binary search.
- If a book is not found for a given ISBN or ID, the function should return **`NULL`**.
- The **`getBookErrorCode`** function should provide insight into why a search may have failed, such as:
    - Invalid catalog passed (uninitialized or freed)
    - Book not found
    - Invalid search value (e.g., negative ISBN or ID)

Notes:

- Titles should be dynamically allocated (malloc'ed). You must be careful to **not leak memory,** clean up all allocations in **`freeBookCatalog`**.
- The **`BookCatalog`** struct (which you must define) should encapsulate all the sorted arrays and any necessary metadata to support the functions above.
- Feel free to use helper functions such as **`binarySearchByISBN`** and **`binarySearchByID`** for clarity.
- Test your code thoroughly with different book datasets and ensure all functions behave correctly, including in edge cases like zero books or duplicate IDs.