```
// Implementation for Library

#include "library.h"
#include <array>

Library::Library()
{
  books = new Book[100];
  numOfBooks = 0;
  capacity = 100;
}

Library::Library(int size)
{
        books = new Book[size];
        numOfBooks = 0;
        capacity = size;
}

Library::Library(const Library & rhs) //Copy Constructor - Deep copy
{
        capacity = rhs.capacity;
        numOfBooks = rhs.numOfBooks;
        books = new Book[capacity];
        for (int i = 0; i < rhs.capacity; i++)
                books[i] = rhs.books[i];
}

Library::~Library()
{
  delete [] books;
  books = NULL;
  numOfBooks = 0;
}

void Library::printLibrary()
{
        for (int i = 0; i < numOfBooks; i++)
        {
                books[i].print(cout);
                cout << endl;
        }
}

int Library::getCapacity()
{
        return capacity;
}
```

```
bool Library::searchBook(int s)
{
	for (int i = 0; i < numOfBooks; i++)
	{
		if (s == books[i].getISBN())
		{
			return true;
		}
	}
	return false;
}

void Library::insertBook(Book * b)
{
	books[numOfBooks] = * b;
	numOfBooks++;
}

void Library::deleteBook(int s)
{
	int index = 0;
	int i = -1;
	while (books[i].getISBN() != s)
	{
		i++;
		if (i > numOfBooks)
		{
			i = -1;
			break;
		}
	}
	index = i;
	if (index == -1)
		return;

	books[index] = books[numOfBooks - 1];
	numOfBooks--;
}

int Library::numBooksInLibrary()
{
	return numOfBooks;
}
```