

### Troubleshooting:

If the given project (e.g. Q1) runs with error, you need to run "Clean and Build Project" (Shift+F11). If still error, try to rename or copy the project to other one, e.g. from Q1 to Q1X or Q1Y.

### Library Management System

The given files already contain statements to implement a program for managing a library system. The structure of the main classes is as follows:

- **Class Book:** contains information about a Book object, including title, copies, and price.
- **Class Node:** includes a Book object and left/right pointers for BST implementation.
- **Class QueueNode:** includes a Book object and a next pointer for queue implementation.
- **Class BookBST:** is a binary search tree that manages all books in the library catalog.
- **Class BorrowQueue:** is a queue structure (implemented as a linked list), where the component data of the Nodes only contains the title and copies. This means that the corresponding request is to borrow a book with a specific title and number of copies.
- **Class LibrarySystem:** is the main class of the program, containing a BookBST used to store data on the books currently available in the library and a BorrowQueue corresponding to a queue of people wanting to borrow books.

Students are required to carefully read the provided code segments to fully understand the relationships between the classes and the functions within each class. The specific task of the test is to execute the following requirements:

#### a. f1(): (2.5 marks)

To complete the requirement f1, students need to fulfill two specific tasks: implement the function insert() in the BookBST structure and the function enqueue() in the BorrowQueue structure. The expected output used to test your code are as follows:

Book Catalog (BST Preorder Traversal):

(Java,10,50.0) (C++,8,45.0) (Algorithms,5,70.0) (Python,15,40.0) (SQL,7,35.0)

Borrow Requests (Queue):

(Java,2) (Python,3) (Ruby,1) (SQL,5) (C++,2)

#### b. f2(): (2.5 marks)

To complete the requirement f2, students need to perform two specific tasks: implement the dequeue() function of the BorrowQueue structure, then use the returned result to execute the borrowing action for the f2 function. The borrowing action consists of two steps:

- First, search for the book (title) to be borrowed. If found, proceed to step 2.

- If the number of copies available in the BookBST is not less than the number to be borrowed, perform the borrowing action: update the remaining copies in the BookBST.

The expected output used to test your code are as follows:

Book Catalog (BST Preorder Traversal):

(Java,10,50.0) (C++,8,45.0) (Algorithms,5,70.0) (Python,15,40.0) (SQL,7,35.0)

Borrow Requests (Queue):

(Java,2) (Python,3) (Ruby,1) (SQL,5) (C++,2)

Book Catalog (BST Preorder Traversal):

(Java,8,50.0) (C++,8,45.0) (Algorithms,5,70.0) (Python,15,40.0) (SQL,7,35.0)

Borrow Requests (Queue):

(Python,3) (Ruby,1) (SQL,5) (C++,2)

**Explanation:** Initially, the BorrowQueue contains 5 elements corresponding to 5 book borrowing requests. After performing deQueue to remove the first request (Java,2), the BorrowQueue will only have 4 elements. When executing the borrowing action, since the Java book has 10 copies and the request is for 2 copies, after completing the borrowing action, the remaining copies will be 8.

**c. f3(): (2.5 marks)**

Perform the pair of operations deQueue() and borrowBook() for all book borrowing requests in the BorrowQueue. The expected output used to test your code are as follows:

Book Catalog (BST Preorder Traversal):

(Java,10,50.0) (C++,8,45.0) (Algorithms,5,70.0) (Python,15,40.0) (SQL,7,35.0)

Borrow Requests (Queue):

(Java,2) (Python,3) (Ruby,1) (SQL,5) (C++,2)

Book Catalog (BST Preorder Traversal):

(Java,8,50.0) (C++,6,45.0) (Algorithms,5,70.0) (Python,12,40.0) (SQL,2,35.0)

Borrow Requests (Queue):

Empty

**Explanation:** Since the borrowing action is executed for all requests in the BorrowQueue, at the end, the BorrowQueue will be empty. For the requested books, Java, Python, SQL, and C++ will be updated because the requested copies are available. However, Ruby does not exist in the BookBST, so this borrowing request cannot be performed.

**d. f4(): (2.5 marks)**

Perform the pair of actions, `deQueue()` and `borrowBook()`, for all the books in the `BorrowQueue` and calculate the total value of borrowed books. The value is equal to the multiplication of the copies borrowed and the price of each book. The expected output used to test your code are as follows:

Book Catalog (BST Preorder Traversal):

(Java,10,50.0) (C++,8,45.0) (Algorithms,5,70.0) (Python,15,40.0) (SQL,7,35.0)

Borrow Requests (Queue):

(Java,2) (Python,3) (Ruby,1) (SQL,5) (C++,2)

Total Value of Borrowed Books: 505.0

**Explanation:** There are 5 book borrowing requests in the `BorrowQueue`, where Ruby cannot be borrowed. The 4 books that can be borrowed are Java, Python, SQL, and C++. The number of copies of Java, Python, SQL, and C++ in the `BorrowQueue` are 2, 3, 5, and 2. The prices in the `BookBST` are 50.0, 40.0, 35.0, and 45.0, respectively. Therefore, the total value is calculated as:

$\text{Value} = 2 \times 50.0 + 3 \times 40.0 + 5 \times 35.0 + 2 \times 45.0 = 100.0 + 120.0 + 175.0 + 90.0 = 505.0$