Binary Search Symbol Table

1. ***Implement the API using BinarySearchST:*** Implement the following API using BinarySearchST.

void **put** (Key k, Value v) inserts key and a value at appropriate position in the arrays.

boolean **contains**(Key k)   return true if the given key is in the symboltable.

Value **get**(Key k) return value paired with Key.

Key **max**() return largest key

Key **floor**(Key key) return largest key less than or equal to key

int **rank**(Key key) return number of keys less than key

void **deleteMin**() delete smallest key

Iterable<Key> **keys**() return all keys, in sorted order

**Note:**

* While inserting the values in to the symbol table, take the default values from 0 to n-1 where n is the number of keys.
* Print the maximum key when the **max** method is invoked.
* Print the largest key less than or equal to the given key when the **floor** method is invoked.
* Print the number of keys less than the given key when the **rank** method is invoked.
* Print nothing when the **deleteMin** method is invoked.
* Print true / false for the given key when **contains** method is invoked.
* Print the key and value pair separated by a space in each line when the **keys** method is invoked.
* Print the value associated with the given key when **get** method is invoked. Print null if there is no value associated with the key or key not found.
* Check for the test cases given in the folder.

**Test cases: Your code will be tested by unit testing. So follow the given API strictly.**