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 min() return smallest 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

**Input Format:**

* The first line of the input contains the keys that are separated by spaces.
* From the second line onwards, there are API method calls with respect to the method prototype that are separated by spaces.

**Output Format:**

* Print the output for the respective method as shown in the test cases.

**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.

**Sample Input #1:**

**S E A R C H E X A M P L E**

**max**

**floor R**

**rank B**

**deleteMin**

**contains C**

**keys**

**get A**

**Sample Output #1:**

**X**

**R**

**1**

**true**

**C 4**

**E 12**

**H 5**

**L 11**

**M 9**

**P 10**

**R 3**

**S 0**

**X 7**

**null**