

class name: <code>Date</code> method signature: <code>public boolean isValid() {}</code> //check if a given date is a valid calendar date			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	The valid range for the month shall be ≥ 1 and ≤ 12 . The method shall return false for a month value outside the valid range.	<ul style="list-style-type: none"> Create two instances of Date with a valid day and year but one with a month > 12 and one with a month < 1. test data: "15/21/2015" test data: "0/21/2015" 	false
2	The valid range for the day for the month of January, March, May, July, August, October, and December shall be ≥ 1 and ≤ 31 . The method shall return false for a date value outside this valid range.	<ul style="list-style-type: none"> Create two instances of Date with the month = 1, a valid year, but one with the day > 31, and one with the day < 1. test data: "1/32/1997" test data: "1/0/1997" 	false
3	The valid range for the day for the month of April, June, September, and November shall be ≥ 1 and ≤ 30 . The method shall return false for a date value outside this valid range.	<ul style="list-style-type: none"> Create two instances of Date with the month = 6, a valid year, but one with the day > 30, and one with the day < 1. test data: "6/31/2000" test data: "6/0/2000" 	false
4	The number of days in February for a non-leap year shall be 28. The method shall return false if the date given has > 28 days for a non-leap year.	<ul style="list-style-type: none"> Create an instance of Date with February, month = 2, day > 28, and the year is a non-leap year test data: "2/29/2005" 	false
5	The number of days in February for a leap year shall be 29. The method shall return false if the date given has > 29 days for a leap year.	<ul style="list-style-type: none"> Create an instance of Date with the month = 2, day > 29, and the year is a leap year test data: "2/30/2000" 	false
6	The valid range for the day for the month of January, March, May, July, August, October, and December shall be ≥ 1 and ≤ 31 . The valid range for the month shall be ≥ 1 and ≤ 12 . The method shall return true for a date value and month value in this valid range.	<ul style="list-style-type: none"> Create an instance of Date with January, month = 1, the day being between 1 and 31 inclusively and a valid year. test data: "1/27/2009" 	true
7	Return true if there are ≤ 29 days in a leap year.	<ul style="list-style-type: none"> Create an instance of Date with a year being a leap year, the day being ≤ 29, and with a valid month. test data: "2/29/2000" 	true

class name: Artist

method signature: `public int compareTo() {}` //compares two given artists, first based on their name and then their date of birth.

Test Case #	Requirement	Test description and Input Data	Expected result/output
1	When the first artist's name is "less than" or comes before in the lexicographic order than the second artist's name, this method returns -1, meaning the first artist will come before the second one, regardless of whether their date of birth is the same or not.	<ul style="list-style-type: none">Create two instances of Artist with the first one's name coming before the second one's name in the lexicographic order, with the same date of birth.test data: "Abraham Lincoln" "11/12/2004" and "John Wang" "11/12/2004"	-1
2	When two artists' names are the same, this method then compares their date of birth. This method returns -1 when the current/first artist's date of birth comes before the other/second artist's date of birth, meaning the first artist will come before the second one.	<ul style="list-style-type: none">Create two instances of Artist with the same name, but the first one's date of birth comes before the second one's date of birth.test data: "Mark Sandberg" "1/26/1990" and "Mark Sandberg" "8/11/2016"	-1
3	When the first artist's name is "greater" or comes after in the lexicographic order than the second artist's name, this method returns 1, meaning the first artist will come after the second one regardless of whether their date of birth is the same or not.	<ul style="list-style-type: none">Create two instances of Artist with the first one's name coming after the second one's name in the lexicographic order, with different dates of birth.test data: "Vivek Manthri" "7/4/1990" and "Ved Patel" "10/15/2003"	1
4	When two artists' names are the same, this method then compares their date of birth. This method returns 1 when the current/first artist's date of birth comes after the other/second artist's date of birth, meaning the first artist will come after the second one.	<ul style="list-style-type: none">Create two instances of Artist with the same name, but the first one's date of birth comes after the second one's date of birth.test data: "Zuck Riggy" "12/15/1998" and "Zuck Riggy" "12/4/1998"	1
5	When the two artists' names and dates of birth are the same, this method returns 0.	<ul style="list-style-type: none">Create two instances of Artist with the same name, and the same date of birthtest data: "Bob Builder" "6/6/2000" and "Bob Builder" "6/6/2000"	0