

<b>Class Name:</b> Date <b>Method Signature:</b> public boolean isValid()			
Test Case #	Requirement	Test Description and Input Data	Expected Result/Output
1	<b>Invalid Date:</b> The method shall return false with a date that contains a year before the year 1900	<ul style="list-style-type: none"> <li>Create an instance of a date that contains a year of 1800, as the year 1800 is too far back into the past</li> <li>Test Input: "02/30/1800"</li> </ul>	Expected Output: <b>False</b>  Actual Output: <b>False</b>
2	<b>Invalid Date:</b> The method shall return false with a date that contains an invalid day value due to the corresponding month not having that day value in their day range	<ul style="list-style-type: none"> <li>Create an instance of an invalid date, in this example, it would be April 31st, as the month of April does not have a 31st day</li> <li>Test Input: "04/31/2023"</li> </ul>	Expected Output: <b>False</b>  Actual Output: <b>False</b>
3	<b>Invalid Date:</b> The method shall return false with a date that contains an invalid month value	<ul style="list-style-type: none"> <li>Create an instance of a date that contains a month value of 13, as there is no 13th month in our calendar</li> <li>Test Input: "13/15/2023"</li> </ul>	Expected Output: <b>False</b>  Actual Output: <b>False</b>
4	<b>Invalid Date:</b> The method shall return false with a date that contains an invalid day value	<ul style="list-style-type: none"> <li>Create an instance of a date that contains a day value of 0, as there is no 0th day in our calendar</li> <li>Test Input: "12/00/2023"</li> </ul>	Expected Output: <b>False</b>  Actual Output: <b>False</b>
5	<b>Invalid Date:</b> The method shall return false with a date that contains a 29th day in February during a non-leap year	<ul style="list-style-type: none"> <li>Create an instance of a date that includes the 29th day of February when it is not a leap year, in this case, it would be 2023, as the year 2023 is a non-leap year and does not contain the 29th day in February</li> </ul>	Expected Output: <b>False</b>  Actual Output: <b>False</b>

		<ul style="list-style-type: none"> <li>Test Input: "02/29/2023"</li> </ul>	
6	<b>Valid Date:</b> The method shall return true with a date that contains a 29th day in February during a leap year	<ul style="list-style-type: none"> <li>Create an instance of a date that includes the 29th day of February when it is a leap year, in this case, it would be 2024, as the year 2024 is a leap year and contains the 29th day in February</li> <li>Test Input: "02/29/2024"</li> </ul>	Expected Output: <b>True</b>  Actual Output: <b>True</b>
7	<b>Valid Date:</b> The method shall return true with a date that obeys the edge cases provided in the project requirements which defines a valid calendar date	<ul style="list-style-type: none"> <li>Create an instance of a date that is a valid calendar date given that it has the plausible month, day, and year values. In this case, it would be the 31st day of December during 2023.</li> <li>Test Input: "12/31/2023"</li> </ul>	Expected Output: <b>True</b>  Actual Output: <b>True</b>

<b>Class Name:</b> Profile <b>Method Signature:</b> public int compareTo()			
Test Case #	Requirement	Test Description and Input Data	Expected Result/Output
1	<b>Lexicographically Greater:</b> The method shall return 1 when the last name is lexicographically greater than the compared last name	<ul style="list-style-type: none"> <li>Create an instance of a profile where the last name is different and lexicographically greater from the compared last name</li> <li>Test Input: "John Doe 01/01/2000" and "John Anderson 01/01/2000"</li> </ul>	Expected Output: <b>1</b>  Actual Output: <b>1</b>
2	<b>Lexicographically Greater:</b> The method shall return 1 when the first name is	<ul style="list-style-type: none"> <li>Create an instance of a profile where the first name is different and lexicographically greater from the compared first</li> </ul>	Expected Output: <b>1</b>  Actual Output: <b>1</b>

	lexicographically greater than the compared first name	name <ul style="list-style-type: none"> <li>Test Input: “John Doe 01/01/2000” and “Jane Doe 01/01/2000”</li> </ul>	
3	<b>Lexicographically Greater:</b> The method shall return 1 when the date of birth is later than the compared date of birth, as this makes the date lexicographically greater	<ul style="list-style-type: none"> <li>Create an instance of a profile where the date of birth is different and later so it is lexicographically greater from the compared date of birth</li> <li>Test Input: “John Doe 05/01/2000” and “John Doe 01/01/2000”</li> </ul>	Expected Output: <b>1</b>  Actual Output: <b>1</b>
4	<b>Lexicographically Lesser:</b> The method shall return -1 when the last name is lexicographically lesser than the compared last name	<ul style="list-style-type: none"> <li>Create an instance of a profile where the last name is different and lexicographically lesser from the compared last name</li> <li>Test Input: “John Anderson 01/01/2000” and “John Doe 01/01/2000”</li> </ul>	Expected Output: <b>-1</b>  Actual Output: <b>-1</b>
5	<b>Lexicographically Lesser:</b> The method shall return -1 when the first name is lexicographically lesser than the compared first name	<ul style="list-style-type: none"> <li>Create an instance of a profile where the first name is different and lexicographically lesser from the compared first name</li> <li>Test Input: “Jane Doe 01/01/2000” and “John Doe 01/01/2000”</li> </ul>	Expected Output: <b>-1</b>  Actual Output: <b>-1</b>
6	<b>Lexicographically Lesser:</b> The method shall return -1 when the date of birth is earlier than the compared date of birth, as this makes the date lexicographically lesser	<ul style="list-style-type: none"> <li>Create an instance of a profile where the date of birth is different and earlier so it is lexicographically lesser from the compared date of birth</li> <li>Test Input: “John Doe 01/01/2000” and “John Doe 05/01/2000”</li> </ul>	Expected Output: <b>-1</b>  Actual Output: <b>-1</b>

7	<b>Lexicographically Equal:</b> The method shall return 0 if the first name, last name, and date of birth are equal	<ul style="list-style-type: none"><li>• Create an instance of a profile where the date of birth is completely identical to the compared date of birth, which makes it lexicographically equal</li><li>• Test Input: “John Doe 01/01/2000” and “John Doe 01/01/2000”</li></ul>	Expected Output: <b>0</b>  Actual Output: <b>0</b>
---	--	---	--