|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case #** | **Purpose of Test Case** | **Input Data** | **Expected Output** |
| 1 | Test the .isValid() method for valid dates in the current month,day, and year(at the time of testing).   * Case 1: is today's date, returns true * Case 2: is tomorrow's date, returns false * Case 3: is yesterday's date, returns true | * Case 1: “2/9/2021” * Case 2: “2/10/2021” * Case 3: “2/8/2021 | * Case 1 returns true * Case 2 returns false * Case 3 returns true |
| 2 | Test the .isValid() method for valid year ranges, keeping everything else constant and only changing year.   * Case 1: a year before 1900, returns false * Case 2: a date on 1900, returns true * Case 3: a year in the future, returns false | * Case 1: “1/10/1899” * Case 2: “1/10/1900” * Case 3: “1/10/2022” | * Case 1 returns false * Case 2 returns true * Case 3 returns false |
| 3 | Test the .isValid() method for valid month/day ranges while using years that are in valid year ranges.   * Case 1: a month that is too large, returns false * Case 2: a month that is barely too large, returns false * Case 3: a month that is too small, returns false * Case 4: a day that is too small, returns false * Case 5: a day that is too large, returns false * Case 6: a day that is barely too large, returns false | * Case 1: “31/2/2000” * Case 2: “13/2/2020” * Case 3: “0/5/1999” * Case 4: “1/0/1999” * Case 5: “1/50/1999” * Case 6: “2/32/1999” | * Case 1 returns false * Case 2 returns false * Case 3 returns false * Case 4 returns false * Case 5 returns false * Case 6 returns false |
| 4 | Test the .isValid() method for months with their respective number of days, can assume rest are correct after 4 test cases   * Case 1: January has 31 days, returns true * Case 2: April does not have 31 days, returns false * Case 3: May has 31 days, returns true * Case 4: June does not have 31 days, returns false | * Case 1: “1/31/2000” * Case 2: “4/31/2000” * Case 3: “5/31/2000” * Case 4: “6/31/2000” | * Case 1 returns true * Case 2 returns false * Case 3 returns true * Case 4 returns false |
| 5 | Test the .isvalid() method for leap years, for all conditions   * Case 1: Testing the most recent leap year, returns true * Case 2: Testing a non-leap year, returns false * Case 3: Test a leap year that is multiple of 100, but divisible by 400, returns true * Case 4: Test a leap year that that is multiple of 100, but not divisible by 400, returns false * Case 5: Test a leap year after skipped leap year, returns true | * Case 1: “2/29/2020” * Case 2: “2/29/2009” * Case 3: “2/29/2000” * Case 4: “2/29/1900” * Case 5: “2/29/1904” | * Case 1 returns true * Case 2 returns false * Case 3 returns true * Case 4 returns false * Case 5 returns true |
| 6 | Test the .isValid() method for random dates that are valid, to check that it works for valid dates as well   * Case 1,2,3: testing dates that are in valid range, all return true | * Case 1: “1/15/1950” * Case 2: “10/31/1969” * Case 3: “4/20/1969” | * Case 1 returns true * Case 2 returns true * Case 3 returns true |