Dhwanit Gajjar and Harb Chauhan
CS213 Project 3 Test Specification

Class name: `Date`
Method Signature: `public Boolean isValid()`

Test Case #	Requirement	Test description and Input Data	Expected Output
1	The method should return false for any date with a year before 1900.	Create an instance of Date with a valid day and month but with an invalid year < 1900. Test input: "11/21/1800"	false
2	Number of days in February for a non-leap year shall be 28. The method shall return false if the date given has 29 days for a non-leap year.	Create an instance of Date with month = 2, day = 29, and a non-leap year (e.g., 2021). Test input: "2/29/2021"	false
3	The valid range for the month shall be >= 1 and <= 12. The method shall return false for a month outside the valid range.	Create an instance of Date with an invalid month (e.g., 13). Test input: "13/15/2020"	false
4	The valid range for the day shall be >= 1 and <= 31 depending on the month. The method shall return false for a day outside the valid range.	Create an instance of Date with an invalid day (e.g., 32) in a month with only 31 days. Test input: "1/32/2022"	false
5	Valid date within standard range.	Create an instance of Date with month = 12, day = 31, and year = 2022. Test input: "12/31/2022"	true

Class name: `Profile`
Method Signature: `public int compareTo(Profile profile2)`

Test Case #	Requirement	Test description and Input Data	Expected Output
1	The method should return false for any date with a Profile with a greater last name should come first.before 1900.	Profile 1: lname='Tomato', fname='Dr.', dob=1/1/1990 Profile 2: lname='Doofenshmirtz', fname='Dr.', dob=1/1/1990	1
2	Profile with the same last name and a greater first name should come first.	Profile 1: lname='Beef', fname='Jonny', dob=1/1/1990 Profile 2: lname='Beef', fname='Angus', dob=1/1/1990	1
3	Profiles with the same last name, first name, and a later date of birth should come first.	Profile 1: lname='Doe', fname='John', dob=1/1/2090 Profile 2: lname='Doe', fname='John', dob=1/1/1990	1
4	Profiles with the same last name, first name, and date of birth should be considered equal.	Profile 1: lname='Doe', fname='John', dob=1/1/1990 Profile 2: lname='Doe', fname='John', dob=1/1/1990	0
5	Compare profiles where both last and first names differ.	Profile 1: lname='Brown', fname='Charlie', dob=1/1/1990 Profile 2: lname='Green', fname='Alice', dob=1/1/1990	-1 / 1 based on alphabetical order

Class name: `List<E>`
Method Signature:
`- public void add(E e)`
`- public void remove(E e)`

Test Case #	Requirement	Test description and Input Data	Expected Output
1	Add an element to an empty list.	List initially empty. Add element 'A'.	List size = 1; Element 'A' at index 0
2	Add an element to a non-empty list.	List with ['A', 'B']. Add element 'C'.	List size = 3; Element 'C' at index 2
3	Remove an existing element from the list.	List with ['A', 'B', 'C']. Remove element 'B'.	List size = 2; List = ['A', 'C']
4	Attempt to remove a non-existent element.	List with ['A', 'C']. Remove element 'B'.	List size = 2; No change
5	Add elements beyond initial capacity to test automatic growth.	Start with an empty list. Add elements 'A', 'B', 'C', 'D', 'E'.	List size = 5; Capacity expands as needed