Boundary Value Testing – NextDate Problem

NextDate is a function of three variables: month, day, and year. It returns the date of the day after the input date in the form "mm/dd/yyyy" (as a string). The month, day, and year variable values are subject to these conditions (the year range ending in 2012 is arbitrary):

1. $1 <= $ month $ <= 12$
2. $1 <= $ day $ <= 31$
3. $1812 <= $ year $ <= 2012$

If an input value fails any of the conditions 1 – 3, the program throws the exception `InvalidValueException`.

If the input values do not make up a valid date, the program throws the exception `InvalidDateException`.

Normal Boundary Value Test Cases: <u>4*n* + 1</u> test cases for a function of *n* variables

Boundary values {min, min+, nominal, max-, max} for month: {1, 2, 6, 11, 12}
Boundary values {min, min+, nominal, max-, max} for day: {1, 2, 15, 30, 31}
Boundary values {min, min+, nominal, max-, max} for year: {1812, 1813, 1912, 2011, 2012}

Note these contain only <u>**valid**</u> values.

Test Case #	Month	Day	Year	Expected Output
1	1	15	1912	"01/16/1912"
2	2	15	1912	"02/16/1912"
3	6	15	1912	"06/16/1912"
4	11	15	1912	"11/16/1912"
5	12	15	1912	"12/16/1912"
6	6	1	1912	"06/02/1912"
7	6	2	1912	"06/03/1912"
8	6	15	1912	"06/16/1912"
9	6	30	1912	"07/01/1912"
10	6	31	1912	InvalidDateException
11	6	15	1812	"06/16/1812"
12	6	15	1813	"06/16/1813"
13	6	15	1912	"06/16/1912"
14	6	15	2011	"06/16/2011"
15	6	15	2012	"06/16/2012"

Robust Boundary Value Test Cases: <u>6*n* + 1</u> test cases for a function of *n* variables

Boundary values {min-, min, min+, nominal, max-, max, max+} for month: {0, 1, 2, 6, 11, 12, 13}
Boundary values {min-, min, min+, nominal, max-, max, max+} for day: {0, 1, 2, 15, 30, 31, 32}
Boundary values {min-, min, min+, nominal, max-, max, max+} for year: {1811, 1812, 1813, 1912, 2011, 2012, 2013}

Note these contain **<u>valid and invalid</u>** values.

Test Case #	Month	Day	Year	Expected Output
1	0	15	1912	InvalidValueException
2	1	15	1912	"01/16/1912"
3	2	15	1912	"02/16/1912"
4	6	15	1912	"06/16/1912"
5	11	15	1912	"11/16/1912"
6	12	15	1912	"12/16/1912"
7	13	15	1912	InvalidValueException
8	6	0	1912	InvalidValueException
9	6	1	1912	"06/02/1912"
10	6	2	1912	"06/03/1912"
11	6	15	1912	"06/16/1912"
12	6	30	1912	"07/01/1912"
13	6	31	1912	InvalidDateException
14	6	32	1912	InvalidValueException
15	6	15	1811	InvalidValueException
16	6	15	1812	"06/16/1812"
17	6	15	1813	"06/16/1813"
18	6	15	1912	"06/16/1912"
19	6	15	2011	"06/16/2011"
20	6	15	2012	"06/16/2012"
21	6	15	2013	InvalidValueException

Worst-Case Boundary Value Test Cases: 5^n test cases for a function of n variables

Boundary values {min, min+, nominal, max-, max} for month: {1, 2, 6, 11, 12}
Boundary values {min, min+, nominal, max-, max} for day: {1, 2, 15, 30, 31}
Boundary values {min, min+, nominal, max-, max} for year: {1812, 1813, 1912, 2011, 2012}
Note these contain only **valid** values.

Test Case #	Month	Day	Year	Expected Output
1	1	1	1812	
2	1	1	1813	
3	1	1	1912	
4	1	1	2011	
5	1	1	2012	
6	1	2	1812	
7	1	2	1813	
8	1	2	1912	
9	1	2	2011	
10	1	2	2012	
11	1	15	1812	
12	1	15	1813	
13	1	15	1912	
14	1	15	2011	
15	1	15	2012	
16	1	30	1812	
17	1	30	1813	
18	1	30	1912	
19	1	30	2011	
20	1	30	2012	
21	1	31	1812	
22	1	31	1813	
23	1	31	1912	
24	1	31	2011	
25	1	31	2012	
.				
.				
.				
.				
.				
.				
.				
.				
.				
.				
125				

Robust Worst-Case Boundary Value Test Cases: 7^n test cases for a function of *n* variables

Boundary values {min-, min, min+, nominal, max-, max, max+} for month: {0, 1, 2, 6, 11, 12, 13}
Boundary values {min-, min, min+, nominal, max-, max, max+} for day: {0, 1, 2, 15, 30, 31, 32}
Boundary values {min-, min, min+, nominal, max-, max, max+} for year: {1811, 1812, 1813, 1912, 2011, 2012, 2013}

Note these contain **valid and invalid** values.

Test Case #	Month	Day	Year	Expected Output
1	0	0	1811	
2	0	0	1812	
3	0	0	1813	
4	0	0	1912	
5	0	0	2011	
6	0	0	2012	
7	0	0	2013	
8	0	1	1811	
9	0	1	1812	
10	0	1	1813	
11	0	1	1912	
12	0	1	2011	
13	0	1	2012	
14	0	1	2013	
15	0	2	1811	
16	0	2	1812	
17	0	2	1813	
18	0	2	1912	
19	0	2	2011	
20	0	2	2012	
21	0	2	2013	
22	0	15	1811	
23	0	15	1812	
24	0	15	1813	
25	0	15	1912	
26	0	15	2011	
27	0	15	2012	
28	0	15	2013	
29	0	30	1811	
30	0	30	1812	
31	0	30	1813	
32	0	30	1912	
33	0	30	2011	
34	0	30	2012	
35	0	30	2013	

36	0	31	1811	
37	0	31	1812	
38	0	31	1813	
39	0	31	1912	
40	0	31	2011	
41	0	31	2012	
42	0	31	2013	
43	0	32	1811	
44	0	32	1812	
45	0	32	1813	
46	0	32	1912	
47	0	32	2011	
48	0	32	2012	
49	0	32	2013	
.				
.				
.				
.				
.				
.				
.				
.				
343				