IT314: Software Engineering

Lab Session IX – Specification-based Test Case Generation

Group-30

Group Members:

Margi Hingrajia:	201801014
Priyal Raj:	201801106
Yash Patel:	201801134
Pratik Parmar:	201801211
Harshil Goti:	201801130
Maharshi Vaghela:	201801216
Nishant Shah:	201801403
Jaydeep Machhi:	201801452
Manish Khandar:	201801222
Jaykumar Darji:	201801460

Contribution:

1. **Manish, Maharshi, Pratik, Nishant,Yash, Jaydeep, Harshil: Question-1**
2. **Margi, Priyal, Jay : Question-2**

Q1: Consider a program for determining the previous date. Its input is triple of day, month and year with the following ranges 1 <= month <= 12, 1 <= day <= 31, 1900 <= year <= 2015.The possible output dates would be a previous date or an invalid date. Design the equivalence class test cases?

1. **Enlist which set of test cases have been identified using Equivalence Partitioning and Boundary Value Analysis separately.**
   Ans:

Equivalence Classes :

Year :

1. Year less than 1900 is invalid class.
2. Year between 1900 and 2015 (including 1900 and 2015) is valid class.
3. Year greater than 2015 is invalid class.

Month :

1. Month less than 1 is invalid class.
2. Month between 1 and 12 (including 1 and 12) is valid class.
3. Month greater than 12 is invalid class.

Day :

1. For months 1,3,5,7,8,10,12 :
    a. Day less than 1 is invalid class.
    b. Day between 1 and 31 (including 1 and 31) is valid class.
    c. Day greater than 31 is invalid class.
2. For month 4,6,9,11 :
    a. Day less than 1 is invalid class.
    b. Day between 1 and 30 (including 1 and 30) is valid class.
    c. Day greater than 30 is invalid class.
3. For month 2 and leap year :
    a. Day less than 29 is invalid class.
    b. Day between 1 and 29 (including 1 and 29) is valid class.
    c. Day greater than 29 is invalid class.
4. For moth 2 and not leap year :
    a. Day less than 1 is invalid class.
    b. Day between 1 and 28 (including 1 and 28) is valid class.
    c. Day greater than 28 is invalid class.

Test Cases :

Input	Day	Month	Year	Output
01-01-2001	1	1	2001	Valid : 31-12-2000
31-04-2013	31	4	2013	Invalid
31-02-2020	31	2	2020	Invalid
09-12-2031	9	12	2031	Invalid
20-09-2000	20	9	2000	Invalid
02-05-2001	2	5	2001	Valid: 01-05-2001
29-02-2000	29	2	2000	Valid: 29-02-2000
32-02-1999	32	02	1999	Invalid
10-25-1945	10	25	1945	Invalid
12-13-2014	12	13	2014	Invalid
02-02-2005	2	2	2005	Invalid
10-04-1899	10	4	1899	Invalid
12-12-2021	12	12	2021	Invalid
09-07-2000	9	7	2000	Valid: 08-07-2000
01-01-2015	1	1	2015	Valid: 31-12-2014
01-03-2004	1	3	2004	Valid: 29-02-2004
01-01-201	1	1	201	Invalid
22-12-1994	22	12	1994	Valid: 21-12-1994
20-10-2000	20	10	2000	Valid: 19-10-2000
01-03-2002	1	3	2002	Valid: 28-02-2002
12-123-2019	12	123	2019	Invalid
09-05-2001	9	5	2001	Valid: 08-05-2001
01-12-2006	1	12	2006	Valid: 30-11-2006

1	-	-	-	Invalid
27-13-1990	27	13	1990	Invalid
25-07-2001	25	7	2001	Valid: 24-07-2001
01-09-2016	1	9	2016	Invalid
30-02-1999	30	2	1999	Invalid
29-02-1804	29	2	1804	Invalid
29-02-1805	29	2	1805	Invalid
---	-	-	-	Invalid
00-00-2018	0	0	0	Invalid
11-1-180	11	1	180	Invalid
12-13k2014	-	-	-	Invalid
0m-05-2Vb1	-	-	-	Invalid
29-0Wz1885	-	-	-	Invalid
29-02y1900	-	-	-	Invalid
01z01-2001	-	-	-	Invalid

2. **Modify your programs such that it runs on eclipse IDE, and then execute your test suites on the program. While executing your input data in a program, check whether the identified expected outcome (mentioned by you) is correct or not.**
   Ans:

Problem Statement

Previous Date

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a day of the year in the format of DD-MM-YYYY. This day may or may not be valid. Your task is to return the day before the day which you are given. If the given day is invalid then print a message "Invalid" or print the previous date in the same format as DD-MM-YYYY.

Input

Input contains only a single string denoting the day, which may or may not be valid.

Output

Output must contain only a single line. If the give date is not valid. Print **"Invalid Date"** else print the previous date in the DD-MM-YYYY format.

Examples

input
01-01-2015
output
31-12-2014

input
29-0Wz1805
output
Invalid Date

Note

String may contain more than one words and you also need to handle this.

polygon beta

Professional way to prepare programming contest problems

Create tests

Tests (37) Delete Current Create Testset

« Back to standard view

Testset: tests

Test count: 37

Tests preview

#	Input (first 255 bytes)	Answer (first 255 bytes)	Description	Example
1	01-01-2001 Download	31-12-2000 Download		
2	31-04-2013 Download	Invalid Date Download	Wrong Date	
3	31-02-2020 Download	Invalid Date Download	Wrong Date	
4	09-12-2031 Download	Invalid Date Download	Wrong Year	
5	20-09-2000 Download	19-09-2000 Download		
6	02-05-2001 Download	01-05-2001 Download		
7	29-02-2000 Download	28-02-2000 Download		
8	44-12-1999 Download	Invalid Date Download	Wrong Date	
9	10-25-1945 Download	Invalid Date Download	Wrong Month	
10	12-13-2014 Download	Invalid Date Download	Wrong Month	
11	02-02-2005 Download	01-02-2005 Download		
12	10-04-1899 Download	Invalid Date Download	Wrong Year	
13	12-12-2021 Download	Invalid Date Download	Wrong Year	
14	09-07-2000 Download	08-07-2000 Download		
15	01-01-2015 Download	31-12-2014 Download		Y
16	01-03-2004 Download	29-02-2004 Download		
17	01-01-201 Download	Invalid Date Download	Wrong Year	
18	22-12-1994 Download	21-12-1994 Download		
19	20-10-2000 Download	19-10-2000 Download		
20	01-03-2002 Download	28-02-2002 Download		
21	12-123-2019 Download	Invalid Date Download	Wrong Month	
22	09-05-2001 Download	08-05-2001 Download		
23	01-12-2006 Download	30-11-2006 Download		
24	27-13-1990 Download	Invalid Date Download	Wrong Month	
25	25-07-2001 Download	24-07-2001 Download		

25	25-07-2001 Download	24-07-2001 Download		
26	01-09-2016 Download	Invalid Date Download	Wrong Year	
27	30-02-1999 Download	Invalid Date Download	Wrong Date	
28	29-02-1804 Download	Invalid Date Download	Wrong Year	
29	29-02-1805 Download	Invalid Date Download	Wrong Date and Year	
30	00-00-2018 Download	Invalid Date Download	Wrong Date and Month	
31	11-1-180 Download	Invalid Date Download	Wrong Year	
32	29-02-1900 Download	Invalid Date Download	Wrong Date	
33	12-13k2014 Download	Invalid Date Download	Wrong Format	
34	0m-05-2Vb1 Download	Invalid Date Download	Wrong Format	
35	29-0Wz1805 Download	Invalid Date Download	Wrong Format	Y
36	29-02y1900 Download	Invalid Date Download	Wrong Format	
37	01z01-2001 Download	Invalid Date Download	Wrong Format	

Q2: You are testing an e-commerce system that sells products like caps and jackets. The problem is to create functional tests using boundary-value analysis and equivalence class partitioning techniques for the web page that accepts the orders.
The system accepts a five-digit numeric item ID number from 00000 to 99999. The system accepts a quantity to be ordered, from 1 to 99. If the user enters a previously ordered item ID and a 0 quantity to be ordered, that item is removed from the shopping cart. Based on these inputs, the system retrieves the item price, calculates the item total (quantity times item price), and adds the item total to the cart total. Due to limits on credit card orders that can be processed, the maximum cart total is $999.99

Ans:

- The **constraints** for the system are as follows:

1. Item ID should be between 00000-99999.
2. The quantity that can be ordered should be between 1-99.
3. The maximum of the cart shouldn't exceed $999.99.

- **Equivalence class partitioning:**

For Item ID:

1. Between 00000-99999 (both inclusive): Valid
2. Item ID < 00000: Invalid
3. Item ID > 99999: Invalid

For Quantity:

1. Between 1-99: Valid
2. Quantity 0: Valid, item removed from shopping cart
3. Quantity < 0: Invalid
4. Quantity > 99: Invalid

For Cart Total:

1. Between 0-$999.99: Valid
2. Value > $999.99: Invalid

Equivalence Classes:

Digits in ID < 5 Quantity < 1 Cart total < $ 0	Digits in ID = 5 1 ≤ Quantity ≤ 99 0 ≤ Cart total ≤ $ 999.99	Digits in ID > 5 Quantity > 99 Cart total > $ 999.99

- **Item total = quantity * item price**
  - ➢ Item total adds to Cart total

Index	Test cases	Output
1	Item id=22256, quantity=4	Valid (Cart total (Some $))
2	Item id = 999, quantity=5	Invalid (item id)
3	Item id = 123456, quantity= 8	Invalid (item id)
4	item= 12345, quantity = 102	Invalid (quantity)
5	Item id = 45685, quantity = -3	Invalid (quantity)
6	Item id = 98745, quantity = 0	Valid (Item would be remove from the cart)
7	Item id = 65432, quantity =5, cart total = 1000	Invalid (amount)
8	item id = 12365, quantity = 8, cart total = -10$	Invalid (amount)

❖ **Possible Test Cases using Boundary Value Analysis:**

➢ **For Item ID:**
Test for values: **empty, -1, 00000, 99999, 99999+**

➢ **For Quantity:**
Test for values: **-1, 0, 1, 99, 99+**

➢ **For Cart Total:**
Test for values: **-$0.1, $0, $1, $999.99, $1000+**