

Homework 1: Nest Date Function Assignment

Student ID: 632115006

Name: Kantaporn Sriwichai

The next date function:

- The next date function receive the date, month, and year and then return the next date string consist in the given format “dd/MM/yyyy” where dd is then number of date, MM is a number of month, and yyyy is 4 digit of the year
 - However, the leap year, or the year which is divisible by 4, unless the century year. Century year are leap years only if they are multiples of 400
 - And in the leaf year, February contains 29 days
-

Task 1: What are the *equivalence classes* to the next date function?

Task 2: What are the *weak normal* equivalence class test cases?

Task 3: What are the *weak robust* equivalence class test cases?

Task 4: What are the *strong normal* equivalence class test cases?

Task 5: What are the *strong robust* equivalence class test cases?

Task 1: What are the *equivalence classes* to the next date function?

Answer

Assume that the input variables with value:

- Input variables
 - o $1 \leq \text{day} \leq 31$
 - o $1 \leq \text{month} \leq 12$
 - o $1000 \leq \text{year} \leq 9999$

So, the equivalence classes of this function are:

- Valid Equivalence classes
 - o $D1 = \{\text{day}: 1 \leq \text{day} \leq 31\}$
 - o $M1 = \{\text{month}: 1 \leq \text{month} \leq 12\}$
 - o $Y1 = \{\text{year}: 1000 \leq \text{year} \leq 2023\}$
- Invalid Equivalence classes
 - o $D2 = \{\text{day}: \text{day} < 1\}$
 - o $D3 = \{\text{day}: \text{day} > 31\}$
 - o $M2 = \{\text{month}: \text{month} < 1\}$
 - o $M3 = \{\text{month}: \text{month} > 12\}$
 - o $Y2 = \{\text{year}: \text{year} < 1000\}$
 - o $Y3 = \{\text{year}: \text{year} > 9999\}$

```
date.py
C: > Users > saiparn > Desktop > Const and Test > date.py > ...
1 year = int(input("Input a year: "))
2 if (year > 9999):
3     print("Invalid Input")
4     print("Error")
5     exit()
6 elif (year < 1000):
7     print("Invalid Input")
8     print("Error")
9     exit()
10 elif (year % 400 == 0):
11     leap_year = True
12 elif (year % 100 == 0):
13     leap_year = False
14 elif (year % 4 == 0):
15     leap_year = True
16 else:
17     leap_year = False
18
19 month = int(input("Input a month [1-12]: "))
20 if (month < 1):
21     print("Error")
22     exit()
23 elif (month > 12):
24     print("Error")
25     exit()
26 elif month in (1, 3, 5, 7, 8, 10, 12):
27     month_length = 31
28 elif month == 2:
29     if leap_year:
30         month_length = 29
31     else:
32         month_length = 28
33 else:
34     month_length = 30
35
```

```
date.py
C: > Users > saiparn > Desktop > Const and Test > date.py > ...
34 month_length = 30
35
36 day = int(input("Input a day [1-31]: "))
37 if (day < 1):
38     print("Error")
39     exit()
40 elif (day > 31):
41     print("Error")
42     exit()
43 elif day < month_length:
44     day += 1
45 else:
46     day = 1
47     if month == 12:
48         month = 1
49         year += 1
50     else:
51         month += 1
52
53 print("The next date is [dd/mm/yyyy] %d/%d/%d." % (day, month, year))
54
```

Task 2: What are the *weak normal* equivalence class test cases?

Answer: we work with a single fault assumption; only valid values are tested. The useful situation is to partition the 3 inputs:

- Day: 1 through 28, 29, 30, 31
- Month:
 - o those that have 31 days {1,3,5,7,8,10,12}
 - o those that have 30 days {4,6,9,11}
 - o that has less than 30 days or {2}
- Year: leap year and non-leap year between 1000 and 9999

| Weak Normal Equivalence | | | | | |
|-------------------------|-----|-------|------|-----------------|---------------|
| Case ID | Day | Month | Year | Expected Output | Description |
| C1 | 8 | 10 | 1600 | 9/10/1600 | leap year |
| C2 | 30 | 4 | 2400 | 1/5/2400 | leap year |
| C3 | 31 | 2 | 1900 | 1/3/1900 | non-leap year |
| C4 | 29 | 7 | 2500 | 30/7/2500 | non-leap year |

Task 3: What are the *weak robust* equivalence class test cases?

Answer: we work with a single fault assumption; like weak normal but it tests both valid and invalid.

- $31 < \text{days} < 1$
- $12 < \text{months} < 1$
- $9999 < \text{year} < 1000$

| Weak Robust Equivalence | | | | | |
|-------------------------|-----|-------|-------|-----------------|-------------------|
| Case ID | Day | Month | Year | Expected Output | Description |
| C1 | 45 | 5 | 2000 | Error | Invalid day |
| C2 | -5 | 5 | 2000 | Error | Invalid day |
| C3 | 30 | 22 | 2000 | Error | Invalid month |
| C4 | 15 | 0 | 2000 | Error | Invalid month |
| C5 | 20 | 7 | 10000 | Error | Invalid year |
| C6 | 15 | 7 | 999 | Error | Invalid year |
| C7 | 45 | 22 | 10000 | Error | Invalid All Input |
| C8 | -5 | 0 | 999 | Error | Invalid All Input |

Task 4: What are the *strong normal* equivalence class test cases?

Answer: work with multiple fault assumptions; each valid possibility is tested. The useful situation is to partition the 3 inputs:

- Day: 1 through 28, 29, 30, 31
- Month:
 - o those that have 31 days {1,3,5,7,8,10,12}
 - o those that have 30 days {4,6,9,11}
 - o that has less than 30 days or {2}
- Year: leap year and non-leap year between 1000 and 9999

We should have (2 years * 3 months * 4 days) = 24 test cases.

| Strong Normal Equivalence | | | | | |
|---------------------------|-----|-------|------|-----------------|---------------|
| Case ID | Day | Month | Year | Expected Output | Description |
| C1 | 5 | 10 | 1600 | 6/10/1600 | leap year |
| C2 | 30 | 10 | 1600 | 31/10/1600 | leap year |
| C3 | 31 | 10 | 1600 | 1/11/1600 | leap year |
| C4 | 29 | 10 | 1600 | 30/10/1600 | leap year |
| C5 | 5 | 6 | 1600 | 6/6/1600 | leap year |
| C6 | 30 | 6 | 1600 | 1/7/1600 | leap year |
| C7 | 31 | 6 | 1600 | 1/7/1600 | leap year |
| C8 | 29 | 6 | 1600 | 30/6/1600 | leap year |
| C9 | 5 | 2 | 1600 | 6/2/1600 | leap year |
| C10 | 30 | 2 | 1600 | 1/3/1600 | leap year |
| C11 | 31 | 2 | 1600 | 1/3/1600 | leap year |
| C12 | 29 | 2 | 1600 | 1/3/1600 | leap year |
| C13 | 5 | 10 | 1700 | 6/10/1700 | non-leap year |
| C14 | 30 | 10 | 1700 | 31/10/1700 | non-leap year |
| C15 | 31 | 10 | 1700 | 1/11/1700 | non-leap year |
| C16 | 29 | 10 | 1700 | 30/10/1700 | non-leap year |
| C17 | 5 | 6 | 1700 | 6/6/1700 | non-leap year |
| C18 | 30 | 6 | 1700 | 1/7/1700 | non-leap year |
| C19 | 31 | 6 | 1700 | 1/7/1700 | non-leap year |
| C20 | 29 | 6 | 1700 | 30/6/1700 | non-leap year |
| C21 | 5 | 2 | 1700 | 6/2/1700 | non-leap year |
| C22 | 30 | 2 | 1700 | 1/3/1700 | non-leap year |
| C23 | 31 | 2 | 1700 | 1/3/1700 | non-leap year |
| C24 | 29 | 2 | 1700 | 1/3/1700 | non-leap year |

Task 5: What are the *strong robust* equivalence class test cases?

Answer: work with multiple fault assumptions; work with each valid and invalid possibility.

| Strong Robust Equivalence | | | | | |
|---------------------------|-----|-------|-------|-----------------|---------------|
| Case ID | Day | Month | Year | Expected Output | Description |
| C1 | 80 | 45 | 11000 | Error | Invalid Input |
| C2 | 25 | 45 | 11000 | Error | Invalid Input |
| C3 | 80 | 7 | 11000 | Error | Invalid Input |
| C4 | 25 | 7 | 11000 | Error | Invalid Input |
| C5 | 80 | 45 | 2000 | Error | Invalid Input |
| C6 | 25 | 45 | 2000 | Error | Invalid Input |
| C7 | 80 | 7 | 2000 | Error | Invalid Input |
| C8 | 0 | -2 | 2000 | Error | Invalid Input |
| C9 | 15 | -2 | 800 | Error | Invalid Input |
| C10 | 0 | 4 | 800 | Error | Invalid Input |
| C11 | 10 | 4 | 800 | Error | Invalid Input |
| C12 | 0 | -2 | 2000 | Error | Invalid Input |
| C13 | 10 | -2 | 2000 | Error | Invalid Input |
| C14 | 0 | 4 | 2000 | Error | Invalid Input |