**Code implementation**

import sys

import datetime

# get input func

def get\_input(message):

dateIsValid = False

dateFormat = "%m/%d/%Y"

# while loop false until valid date entered

# check to ensure the year is within the range 1900-2099

while (not dateIsValid):

userInput = input(message)

# try parsing the date from the input. If invalid, then it fails & is caught by the

# except block

try:

date = datetime.datetime.strptime(userInput, dateFormat)

# ensure the YYYY is in the range of 1900 - 2099 inclusively

if (date.date().year < 1900 or date.date().year > 2099):

print("Invalid year. Please enter a year between 1900 and 2099 inclusively.")

# if all is valid, then break the while loop & return the value

else:

dateIsValid = True

except:

print("Invalid input, try again.\n")

# return date

return date

# get next date func

def get\_next\_date(date):

# Get the next date

nextDate = date + datetime.timedelta(days=1)

# Convert the string to MM/DD/YYYY format

nextDateString = str(nextDate.date().month) + '/' + str(nextDate.date().day) + '/' + str(nextDate.date().year)

# Return the string

return nextDateString

# Main

def main():

# get date MM/DD/YYYY between 1900-2099

dateInput = get\_input("Enter a date in the format MM/DD/YYYY: ")

# get next date

nextDate = get\_next\_date(dateInput)

# print the next date

print(nextDate)

# Check for the system version.

# Python 3 is required for the input

# If using python 3, then call the main function

if (sys.version\_info > (3, 0)):

main()

# Otherwise fail gracefully

else:

print("You need python3 for this")

**Input Domain and relevant equivalence partitions**

Input Domain / condition: 01/01/1900 <= DATE <= 12/31/2099

Since only 1 condition, we can derive three Equivalence classes:

EC1 (partition 1): DATE < 01/01/1900

EC2 (partition 2): 01/01/1900 <= DATE <= 12/31/2099

EC3 (partition 3): DATE > 12/31/2099

**Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case # | Description | Date | Expected Output | Actual Output | Result  (Pass/Fail) |
| 1 | Lower Boundary of Partition 1 | 01/01/0001 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 2 | Random Value of Partition 1 | 04/21/1256 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 3 | Upper Boundary of Partition 1 | 12/31/1899 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 4 | Lower Boundary of Partition 2 | 01/01/1990 | 01/02/1990 | 01/02/1990 | Pass |
| 5 | Random Value of Partition 2 | 06/18/2056 | 06/18/2056 | 06/18/2056 | Pass |
| 6 | Upper Boundary of Partition 2 | 12/31/2099 | 01/01/2100 | 01/01/2100 | Pass |
| 7 | Lower Boundary of Partition 3 | 01/01/2100 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 8 | Random Value of Partition 3 | 10/21/2500 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 9 | Upper Boundary of Partition 3 | 12/31/9999 | Invalid year. Please enter a year between 1900 and 2099 inclusively | Invalid year. Please enter a year between 1900 and 2099 inclusively | Pass |
| 10 | Non-date Input | test | Invalid input, try again | Invalid input, try again | Pass |