**Assignment ONE**

Name: Linh Van Ha (116592171)

Subject: OPS435 SAA

Professor: Raymond Chan

Due Date: February 22nd, 2019

Date Submitted: February 15th, 2019

1. **Functions Explanation**
2. **Usage**

Usage function is to check how many arguments user enters and return the proper usage of the script

* + If user enter the number of arguments not like 3 or 4, it will show the instruction what is the proper way to run the command

Examples: Python3 a1\_vlha.py 2 4 5 3

a1\_vlha.py [--step] YYYYMMDD +/-n

1. **Leap\_year**

Leaf\_year functions accepts 1 argument and check if the year given is leap year or not

Leaf\_year(year\_value) => int

Leaf\_year() function take a valid year which will be pass from validate function in “YYYY” format and return True if the given year is leap year and return false if the given year is not leap year

Examples

leap\_year(2018) => False

Leap\_year(2020) => True

(END)

1. **Day\_in\_mon**

* Day\_in\_mon function accepts 1 argument and get the result from leap\_year to decide the maximum day of Februay

Day\_in\_month(year\_value) => int

Function day\_in\_month take a valid year in “YYYY” format and pass to leap\_year function to get the maximum days in February and return a dictionary with keys are all the months in a year and the values are all the maximum day of each month.

Examples

Day\_in\_mon(2018)

* mon\_max = { 1:31, 2:28, 3:31, 4:30, 5:31, 6:30, 7:31, 8:31, 9:30, 10:31, 11:30, 12:31}

Day\_in\_mon(2020)

* mon\_max = { 1:31, 2:29, 3:31, 4:30, 5:31, 6:30, 7:31, 8:31, 9:30, 10:31, 11:30, 12:31}

**(**END**)**

1. **Validate**

* Validate function will check the input from user is in correct format or not and return the error if the date given is not valid.

Validate(var1) => str

Validate() takes a value from argument in “YYYYMMDD” format and return the error if the date is not valid

Eg:

Validate(‘2018050332’) => Error: wrong date entered

Validate(‘20181502’) => Error: wrong month entered

Validate(‘20181132’) => Error: wrong day entered

(END)

1. **Tomorrow**

* Tomorrow function accepts 1 argument and calculate the next day of the given date input

tomorrow(today) -> str

tomorrow() takes a valid date string in 'YYYYMMDD' format and return a

date string for the next day in 'YYYYMMDD' format.

Examples. tomorrow('20171231') -> '20180101'

tomorrow('20180131') -> '20180201'

tomorrow('20180228') -> '20180301'

(END)

1. **Yesterday**

-Yesterday function accepts 1 argument and calculate the previous day of the given date input

yesterday(today) -> str

yesterday() takes a valid date string in 'YYYYMMDD' format and return a

date string for the next day in 'YYYYMMDD' format.

Examples. yesterday('20180101') -> '20171231'

yesterday('20180501') -> '20180431'

yesterday('20180228') -> '20180227'

(END)

1. **DBDA**

* Function dbda() function accepts 2 argument and return either before or after or the number of days between according to the value of the given interger in the same format
* Dbda(value1,value2) => str

Dbda takes 2 valid date string in “YYYYMMDD” for value 1,

Value2 can be either in string format “YYYYMMDD” or “+/-number”

* If value2 is in “YYYYMMDD” format, it will return the number of days between value 1 and value 2
  + Eg:

Dbda(‘20180523,20170523’) => 365

Dbda(‘20180523,20170923’) => 365

* If value2 is like (+/- number), it will return the date before of after according to the give integer of the value2
  + Eg:

Dbda(‘20180101’,’5’) => 20180106

Dbda(‘20180510’,’-5’) => 20180505