**Question:**

A function called print calendar with two integer parameters, yr and mon. This function should first check for the validity of the parameters; that is, yr should be greater than or equal to 1754, and mon must be an integer between 1 and 12 (inclusive). If the input is invalid, the function prints an error message and returns. If the input is valid, the function should print the calendar for month mon in year yr in a neatly formatted manner (see sample runs).

**Parrt-I:**

 Use the wkday on first function to figure out the weekday at which to start printing the dates:

**Answer:**

def wkday\_on\_first(yr,mon):

days=['Sun','Mon','Tue','Wed','Thurs','Fri','Sat']

count=0

for year in range(1754,yr):

count+=yeardays(year)

for month in range(1,mon):

count+=monthdays(yr,month)

return days[count%7]

**Parrt-I:**

 Use the monthdays function to figure out how many dates to print:

**Answer:**

def monthdays(yr,mon):

thirty\_one=[1,3,5,7,8,10,12]

thirty=[4,6,9,11]

if mon in thirty\_one:

return 31

elif mon in thirty:

return 30

else:

if is\_leap(yr):

return 29

else:

return 28

**To print the whole calendar in neet format:**

**Code:**

import calendar

def is\_leap(yr):

return yr%4==0 or (yr%100 and yr%4==0)

def monthdays(yr,mon):

thirty\_one=[1,3,5,7,8,10,12]

thirty=[4,6,9,11]

if mon in thirty\_one:

return 31

elif mon in thirty:

return 30

else:

if is\_leap(yr):

return 29

else:

return 28

def yeardays(yr):

return 365+is\_leap(yr)

def wkday\_on\_first(yr,mon):

days=['Sun','Mon','Tue','Wed','Thurs','Fri','Sat']

count=0

for year in range(1754,yr):

count+=yeardays(year)

for month in range(1,mon):

count+=monthdays(yr,month)

return days[count%7]

def print\_calender(yr,mon):

if yr>1754:

print(calendar.month(yr,mon))

else:

print("year should be after 1754 !..Try again...")

yr=input("enter yr")

mon=input("enter mon")

print(is\_leap(yr))

monthdays(yr,mon)

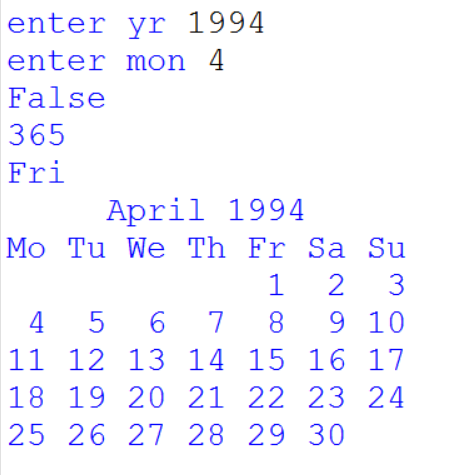
print(yeardays(yr))

print (wkday\_on\_first(yr,mon))

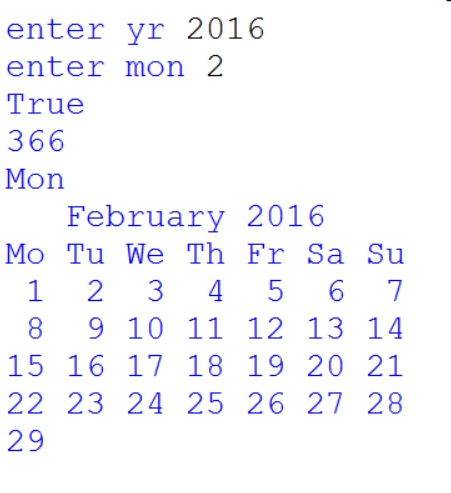
print\_calender(yr,mon)

**Output:**

**1.**

****

**2.**

****