In [1]:

```
from csv import reader
opened_files = open("./glo/moma.csv", encoding="UTF-8")
read_file = reader(opened_files)
list_moma = list(read_file)
list_moma_header = list_moma[0]
list_moma = list_moma[1:]
```

In [2]:

```
list_moma_header
```

Out[2]:

```
['Title',
 'Artist',
 'Nationality',
 'BeginDate',
 'EndDate',
 'Gender',
 'Date',
 'Department']
```

In [6]:

```
list_moma[0][6]
```

Out[6]:

```
'1986'
```

In [7]:

```
list_moma_header[2] == "Nationality"
```

Out[7]:

```
True
```

In [8]:

```
list_1_moma = []
for checks in list_moma:
    nationality = checks[2]
    nationality = nationality.replace("(", "")
    nationality = nationality.replace(")", "")
    if not nationality:
        nationality = "Nationality Unknown"
    checks[2] = nationality
    list_1_moma.append(checks)
```

In [9]:

```
list_1_moma[0][6]
```

Out[9]:

```
'1986'
```

In [10]:

```
list_2_moma = []
for checks in list_1_moma:
    gender = checks[5]
    gender = gender.replace("(", "")
    gender = gender.replace(")", "")
    gender = gender.title()
    if not gender:
        gender = "Gender Unknown/Other"
    checks[5] = gender
    list_2_moma.append(checks)
```

In [11]:

```
list_2_moma[0][6]
```

Out[11]:

```
'1986'
```

In [12]:

```
list_3_moma = []
for checks in list_2_moma:
    birth_date = checks[3]
    death_date = checks[4]
    if birth_date != "":
        birth_date = abs(int(birth_date))
        #death_date = abs(int(death_date))
        birth_date = abs(int(birth_date))
        #death_date = abs(int(death_date))
    checks[3] = (birth_date)
    list_3_moma.append(checks)
```

In [13]:

```
list_3_moma[0][6]
```

Out[13]:

```
'1986'
```

In [ ]:

```
list_4_moma = []
for checks in list_3_moma:
    #birth_date = checks[3]
    death_date = checks[4]
    if death_date != "":
        #birth_date = abs(int(birth_date))
        death_date = abs(int(death_date))
        #birth_date = abs(int(birth_date))
        death_date = abs(int(death_date))
    checks[4] = (death_date)
    list_4_moma.append(checks)
```

In [ ]:

```
list_4_moma[0][6]
```

In [ ]:

```
list_4_moma[780][6]
```

In [ ]:

```
length = 0
for checks in list_4_moma:
    death_date = checks[4]
    if death_date == "":
        length += 1
```

In [ ]:

```
strings = ["good!", "morn?ing", "good?!", "morniZZZZng"]
bad_chars = ["!", "?", "Z"]
def function(string):
    for char in bad_chars:
        string = string.replace(char, "")
    return string
```

In [ ]:

```
list_4_moma
```

In [ ]:

```
cleaned_string = []
for each in strings:
    each = function(each)
    cleaned_string.append(each)
print(cleaned_string)
```

In [ ]:

```
not_good_char = ["(",")","c","C",".","s","'", " "]
def extractor(string):
    for check in not_good_char:
        string = string.replace(check, "")
    return string
```

In [ ]:

```
list_5_moma = []
for checks in list_4_moma:
    date = checks[6]
    date = extractor(date)
    checks[6] = date
    list_5_moma.append(checks)
```

In [ ]:

```
list_5_moma
```

In [ ]:

```
def add(lists):
    sumation = 0
    for each in lists:
        sumation += int(each)
        round_summation = round(sumation / 2)
    return round_summation
```

In [ ]:

```
 add(three_peat)
```

In [ ]:

```
lists = ['1937', '1938']
```

In [ ]:

```
three_peat = "1991-1993"
three_peat = three_peat.split("-")
```

In [ ]:

```
keep = ["(",")","c","C",".","s","'", " "]
for checks in list_5_moma:
    date = checks[6]
```

In [ ]:

```
clean_and_convert("1924-1824")
```

In [ ]:

```
def use(lists):
    sumation = 0
    for each in lists:
        sumation += int(each)
        round_summation = round(sumation / 2)
    return round_summation
```

In [4]:
```
list_4_moma[0][6]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11244\35163052.py in <module>
----> 1 list_4_moma[0][6]

NameError: name 'list_4_moma' is not defined
```

In [ ]:
```
def use(lists):   #my function
    sumation = 0
    for each in lists:
        if each == "":
            each = int(0)
        sumation += int(each)
        sumation = round(sumation / 2)
    return sumation

def clean_and_convert(datee):
    # check that we don't have an empty string
    if datee != "":
        if "-" in datee:
            # move the rest of the function inside
            # the if statement
            datee = datee.split("-")
            datee = use(datee)
    return datee

list_6_moma = []
for checks in list_4_moma:
    datee = str(checks[6])
    datee = clean_and_convert(datee)
    checks[6] = int(datee)
    list_6_moma.append(checks)
```

In [ ]:
```
list_6_moma
```

In [ ]:
```
for checks in list_5_moma:
    date = checks[6]
    if date == "":
        print(checks)
    print("none")
```

In [ ]:
```
list_5_moma[0][6]
```

In [ ]:
```
#def clean_and_convert(datee):
    # check that we don't have an empty string
    if datee != "":
        # move the rest of the function inside
        # the if statement
        datee = datee.split("(", "")
        datee = datee.split(")", "")
        datee = datee.split("c", "")
        datee = datee.split("C", "")
        datee = datee.split(".", "")
        datee = datee.split(",", "")
        datee = datee.split("'", "")
        datee = datee.split("C", "")
        datee = datee.split(" ", "")
    return datee
```

In [ ]:
```
list_7_moma = []
for checks in list_5_moma:
    datee = checks[6]
    print(datee)
```

In [ ]:
```
list_5_moma[0][6]
```

In [ ]:
```
list_6_moma = []
for checks in list_5_moma:
    #birth_date = checks[3]
    date = checks[6]
    if date != "":
        #birth_date = abs(int(birth_date))
        date = abs(int(date))
        #birth_date = abs(int(birth_date))
        date = abs(int(date))
    checks[6] = date
    list_4_moma.append(checks)
```

In [ ]:
```
def use(lists):
    sumation = 0
    for each in lists:
        if each[0] == "":
            each = int(0)
        sumation += int(each)
        round_summation = round(sumation / 2)
    return round_summation
```

In [ ]:
```
list_5_moma[0][6]
```

In [ ]:

```
list_7_moma = []
for checks in list_5_moma:
    date = checks[6]
    date = date.replace("[", "")
    date = date.replace("]", "")
    checks[6] = int(date)
    list_7_moma.append(checks)
```

In [ ]:

```
type(list_5_moma[0][6])
```

In [ ]:

```
def empty(string):
    if "" != string:
        string = int(string)
    return string
```

In [ ]:

```
def empty(string):
    if "" != string:
        string = int(string)
    return string

age_list = []
for checks in list_5_moma:
    birth = checks[3]
    release = checks[6]
    birth = empty(birth)
    release = empty(release)
    if birth == "":
        age = 0
    else:
        age = release - birth
    age_list.append(age)
```

In [ ]:

```
list_5_moma[3][3]
```

In [ ]:

```
list_5_moma[3][6]
```

In [ ]:

```
age_list
```

In [ ]:

```
final_age_list = []
for checks in age_list:
    if checks > 20:
        final_age_list.append(checks)
    else:
        final_age_list.append("Unknown")
```

In [ ]:

```
not_known = []
for checks in final_age_list:
    if checks == "Unknown":
      not_known.append(checks)
print(len(not_known))
```

In [ ]:
```
print(final_age_list)
```

In [ ]:
```
decade = []
ages = []
for checks in final_age_list:
    slise = str(checks)
    if slise == "Unknown":
        ages.append(slise)
    else:
        slise = slise[:-1]
        slise = slise + "0s"
        checks = slise
        decade.append(checks)
```

In [ ]:
```
decade
```

In [ ]:
```
decade_dictionary = {}
for checks in decade:
    if checks in decade_dictionary:
        decade_dictionary[checks] += 1
    else:
        decade_dictionary[checks] = 1
```

In [ ]:
```
decade_dictionary
```

In [ ]:
```
age_dictionary = {}
for checks in final_age_list:
    if checks in age_dictionary:
        age_dictionary[checks] += 1
    else:
        age_dictionary[checks] = 1
```

In [ ]:
```
age_dictionary
```

In [ ]:
```
list_5_moma
```

In [ ]:
```
artist_name = 'Dress MacLeod from Tartan Sets'
birth_date = 1947
release_date = 1986
print(artist_name, "was born in the year", birth_date, "and published his first b
```

In [ ]:
```
print(artist_name + "was born in the year" + str(birth_date) + "and published his
```

In [ ]:

```
artist_frequency = {}
for checks in list_5_moma:
    artist = checks[0]
    if artist in artist_frequency:
        artist_frequency[artist] += 1
    else:
        artist_frequency[artist] = 1
    #print("{} published {} works".format(artist, artist_frequency[artist]))
    #print(f"{artist} published {artist_frequency[artist]} works")
    #print(artist, "published", artist_frequency[artist], "work")
    print(artist + " " + "published" + " " + str(artist_frequency[artist]) + " "
```

In [ ]:

```
artist_frequency
```

In [ ]:

```
pop_millions = [
    ["China", 1379.302771],
    ["India", 1281.935991],
    ["USA",  326.625791],
    ["Indonesia",  260.580739],
    ["Brazil",  207.353391],
]
```

In [ ]:

```
for pop in pop_millions:
    country = pop[0]
    population = pop[1]
    print("the population of {} is {:.2f} million".format(country, population))
```

In [ ]:

```
gender_dictionary = {}
for checks in list_5_moma:
    gender = checks[5]
    if gender in gender_dictionary:
        gender_dictionary[gender] += 1
    else:
        gender_dictionary[gender] = 1
```

In [ ]:

```
gender_dictionary
```

In [ ]:

```
for checks in gender_dictionary:
    print("the {} gender has {} works".format(checks, gender_dictionary[checks]))
```

In [ ]:

```
l = [1, 2, 3]
s = "string"
d = {"a": 1, "b": 2}
```

In [ ]:

```
print(type(d))
```

In [ ]:

```
import pandas as pd
file_df = pd.read_csv("oop.csv")
```

In [ ]:

```
file_df
```

In [1]:

```
import datetime as dt
from csv import reader
opened_file = open("./glo/whitehouse.csv", encoding="UTF-8")
read_file = reader(opened_file)
list_white_house = list(read_file)
white_house_header = list_white_house[0]
list_white_house = list_white_house[1:]
```

In [2]:

```
white_house_header
```

Out[2]:

```
['name',
 'appt_made_date',
 'appt_start_date',
 'appt_end_date',
 'visitee_namelast',
 'visitee_namefirst',
 'meeting_room',
 'description']
```

In [3]:

```
list_white_house[0][3]
```

Out[3]:

```
'01-06-15 23:59'
```

In [5]:

```
appointment = {}
for checks in list_white_house:
    app_date = checks[3]
    day = app_date[8:10]
    month = app_date[5:7]
    year = app_date[0:4]
    month = int(month)
    year = int(year)
    day = int(day)
    dt_object = dt.datetime(year, month, day)
    dt_format = dt_object.strftime("%B, %Y")
    if dt_format in appointment:
        appointment[dt_format] += 1
    else:
        appointment[dt_format] = 1
appointment
```

Out[5]:

```
{'January, 0015': 1248,
 'February, 0015': 2165,
 'March, 0015': 2262,
 'April, 0015': 4996,
 'May, 0015': 3013,
 'June, 0015': 7743,
 'July, 0015': 2930,
 'August, 0015': 1350,
 'September, 0015': 4416,
 'October, 0015': 3669,
 'November, 0015': 1133,
 'December, 0015': 13029}
```

In [4]:
```
for checks in list_white_house:
    start_date = checks[3]
    if "-" in start_date:
        date, time = start_date.split()
        hr, mn = time.split(":")
        month, day, year = date.split("-")
        hr = int(hr)
        mn = int(mn)
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime(year, month, day, hr, mn)
        dt_format = dt_object.strftime("%Y/%m/%d %H:%M")
        start_date = dt_format
        checks[3] = start_date
        print(checks)
    else:
        date, time = start_date.split()
        hr, mn = time.split(":")
        month, day, year = date.split("/")
        hr = int(hr)
        mn = int(mn)
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime(year, month, day, hr, mn)
        dt_format = dt_object.strftime("%Y/%m/%d %H:%M")
        start_date = dt_format
        checks[3] = start_date
        print(checks)
```

```
['Joshua T. Blanton', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:
59', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Jack T. Gutting', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:5
9', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Bradley T. Guiles', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:
59', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Loryn F. Grieb', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:5
9', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Travis D. Gordon', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:5
9', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Taylor D. Gibbs', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:5
9', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Dameriah A. Smith', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 23:
59', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Dylan S. Hopkinstaylor', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/0
6 23:59', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Joseph S. Barbaria', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06 2
3:59', '', 'potus', 'west wing', 'JointService Military Honor Guard']
['Jonathan L. Buckland', '2014-12-18T00:00:00', '01-06-15 9:30', '0015/01/06
```

In [ ]:
```
now = list_white_house[1][3]
```

In [ ]:
```
this = '01-06-15 23:59'
```

In [ ]:
```
this.split()
```

In [ ]:
```
now.split("-")
#now.split("/")
```

In [ ]:
```
import datetime as dt
time = dt.datetime(2000, 12, 28)
```

In [ ]:
```
print(time)
```

In [ ]:
```
eg_1 = dt.datetime(2000, 1, 1, 21, 26, 2)
print(eg_1)
```

In [ ]:
```
eg = list_white_house[100][3]
```

In [ ]:
```
eg
```

In [ ]:
```
date, time = eg.split()
```

In [ ]:
```
date, time
```

In [ ]:
```
mnth, day, year = date.split("/")
```

In [ ]:
```
hr, mn = time.split(":")
```

In [ ]:
```
mnth = int(mnth)
day = int(day)
year = int(year)
hr = int(hr)
mn = int(mn)
```

In [ ]:
```
dt.datetime(year, mnth, day, hr, mn)
```

In [ ]:
```
len(list_white_house)
```

In [ ]:
```
dt_object = dt.datetime(1984, 1, 24, 23, 59)
day = dt_object.day
month = dt_object.month
year = dt_object.year
dt_string = "{}/{}/{}".format(day, month, year)
print(dt_string)
```

In [ ]:

```
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date = appoint_date.split()
    appoint_date = date, time
    appoint_date = date.split("/"), time.split(":")
    appoint_date = date.split("-"), time.split(":")
    appoint_date = int(year), int(month), int(day), int(hour), int(mn)
    appoint_date_object = dt.datetime(appoint_date[0], appoint_date[1], appoint_
                                     appoint_date[4])
    var = appoint_date_object.strftime("%d/%m/%y")
    print(var)
```

In [ ]:

```
#date_format = "%m-%d-%y %H:%M"
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date = appoint_date.split()
    print(appoint_date)
```

In [ ]:

In [ ]:

```
len(appoint_date)
```

In [ ]:

```
list_white_house[0][3]
```

In [ ]:

```
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date
```

In [ ]:

```
print(list_white_house[577][3])
```

In [ ]:

```
date_1_str = "8/12/1984"
date_1_dt = dt.datetime.strptime(date_1_str, "%m/%d/%Y")
print(type(date_1_dt))
print(date_1_dt)
```

In [ ]:

```
date_format = "%m-%d-%y %H:%M"
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date = dt.datetime.strptime(appoint_date, date_format)
    print(appoint_date)
```

In [ ]:

```
print(date_1_dt)
```

In [ ]:

```
date_1_str = "24/12/1984"
date_1_dt = dt.datetime.strptime(date_1_str, "%d/%m/%Y")
print(type(date_1_dt))
print(date_1_dt)
```

In [ ]:

```
date_format = "%m/%d/%y %H:%M"
this_date = '1/30/15 23:59'
```

In [ ]:

```
now = dt.datetime.strptime(this_date, date_format)
```

In [ ]:

```
this_date = '1/30/15 23:59'
```

In [ ]:

```
now
```

In [ ]:

```
date_format = "%m-%d-%y %H:%M"
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date = dt.datetime.strptime(appoint_date, date_format)
    print(appoint_date)
```

In [ ]:

```
dt_object = dt.datetime(1984, 1, 24, 23, 12)
dt_string = dt_object.strftime("%d/%m/%Y %H:%M:%S")
print(dt_string)
```

In [ ]:

```
ok = dt.datetime(15, 5, 8, 10, 45)
```

In [ ]:

```
list_white_house[0][3]
```

In [ ]:

```
ok
```

In [40]:

```
date_format = "%d/%m/%Y %H:%M"

appt_start_date = "02/02/2018 6:34"
appt_start_date_1 = dt.datetime.strptime(appt_start_date, date_format)
print(appt_start_date_1)

```

```
2018-02-02 06:34:00
```

In [ ]:

```
from datetime import datetime
a = datetime.strptime("3/11/2022", "%m/%d/%Y")
b = datetime.strptime("3/18/2022", "%m/%d/%Y")
a>b
```

In [ ]:

```
a = "2/4/1947", "3/5/1948", "1-2-1947", "3/8/1987", "2-3-1956"
```

In [ ]:
```
for checks in a:
    checks = checks.split()
    checks = date, time
    checks = date.split("/"), time.split(":")
    #appoint_date = date.split("-"), time.split(":")
    checks = int(year), int(month), int(day), int(hour), int(minute)
    appoint_date_object = dt.datetime(checks[0], checks[1], checks[2], checks[3],
                                     checks[4])
    year = appoint_date_object.year
    month = appoint_date_object.month
    day = appoint_date_object.day
    hour = appoint_date_object.hour
    minute = appoint_date_object.minute
    var = "{}/{}/{}".format(day, month, year)
    print(checks)
```

In [ ]:
```
for checks in list_white_house:
    appoint_date = checks[3]
    appoint_date = appoint_date.split()
    date, time = appoint_date
    date = date.split("-")
    time = time.split(":")
    appoint_date = date, time
    print(appoint_date)
```

In [ ]:
```
eg_1 = dt.datetime(2000, 1, 1)
print(eg_1)
```

In [ ]:
```
date_1_str = "12/18/2015 16:30"
date, time = date_1_str.split()
hr, mn = time.split(":")
month, day, year = date.split("/")
hr = int(hr)
mn = int(mn)
month = int(month)
day = int(day)
year = int(year)
dt_object =dt.datetime(year, month, day, hr, mn)
"{}/{}/{}".format(year, month, day)
```

In [ ]:

```
a = "2-4-1947 21:22", "3-5-1948 18:28", "3-8-1987 12:00", "3/8/1987 12:00"
for checks in a:
    if "-" in checks:
        date, time = checks.split()
        hr, mn = time.split(":")
        month, day, year = date.split("-")
        hr = int(hr)
        mn = int(mn)
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime(year, month, day, hr, mn)
        print("{}/{}/{}".format(year, month, day))
    else:
        date, time = checks.split()
        hr, mn = time.split(":")
        month, day, year = date.split("/")
        hr = int(hr)
        mn = int(mn)
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime(year, month, day, hr, mn)
        print("{}/{}/{}".format(year, month, day))
```

In [ ]:

```
a = "2-4-1947", "3-5-1948", "3-8-1987", "3/8/1987"
for checks in a:
    if "-" in checks:
        month, day, year = checks.split("-")
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime.strptime(checks, "%d-%m-%Y")
        print(dt_object)
    else:
        month, day, year = checks.split("/")
        month = int(month)
        day = int(day)
        year = int(year)
        dt_object =dt.datetime.strptime(checks, "%d/%m/%Y")
        print(dt_object)
```

In [ ]:

```
dt_object = dt.datetime(1984, 12, 24, 12, 11)
dt_string = dt_object.strftime("%Y/%m/%d %H:%M")
print(dt_string)
```

In [25]:

```
list_white_house[100][2]
```

Out[25]:

```
'01-06-15 10:00'
```

In [26]:

```
timing_list = []
for checks in list_white_house:
    apt_date = checks[3]
    hour = apt_date[11:13]
    seconds = apt_date[14:16]
    hour = int(hour)
    seconds = int(seconds)
    t_object = dt.time(hour, seconds)
    timing_list.append(t_object)
```

In [20]:

```
timing_list
```

Out[20]:

```
[datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
 datetime.time(23, 59),
```

In [18]:

```
min(timing_list)
```

Out[18]:

```
'23:59'
```

In [32]:

```
dt_object = dt.timedelta(weeks=500)
```

In [33]:

```
print(dt_object)
```

```
3500 days, 0:00:00
```

In [ ]: