Week 4: Assignment 3
Calculate the Office shifts

[*Last Updated on:* ***26th April, 2021, 15:00 Hrs***]

Aim

Max is a hard working IT professional who works at two different offices for ***six days of the week***.

On ***even days***, he works at office **'A'** in a day shift from ***1000 Hrs*** to ***1800 Hrs***.

On ***odd days***, he works at office **'B'** in a night shift from ***1800 Hrs*** to ***0000 Hrs*** or ***midnight***.

Given a list of dates in a CSV file, find out for which office did he work on each date in the list and add the appropriate office name next to the date in the new CSV file.

Also find out how many **hours** does he work for office '**A**' and office '**B**' respectively.

A skeleton code stub is provided a starting point in solving the task.

The code stub provided below is to be used. It calls the functions `readWorkSheet()` , `calculateOfficeHrs()` and `writeOfficeWorkSheet()` . Your task is to complete these functions.

Given

Two files are provided to solve this assigment.

- Skeleton program file: `assignment3.py`
  - The skeleton consists of three functions which you have to modify:
    - `readWorkSheet()`
    - `calculateOfficeHrs()`
    - `writeOfficeWorkSheet()`
- Sample CSV file: `week4_assignment3_sample.csv`
  - The CSV file has 2 columns: `date` and `office_name` .
  - The `date` column contains dates in the `YYYY-MM-DD` format and the `office_name` column is blank.

Procedure

- Open the skeleton program file, **`assignment3.py`** .
- You will notice pre-written comments included in skeleton program for your assistance to solve the assignment.
- Three functions to modify are:
  - `readWorkSheet()`

Function Name	**`readWorkSheet()`**
Purpose	Reads the input CSV file of Work Sheet and creates a mapping of date and office name where he worked. The office name should be **`A`** and **`B`** for even and odd days respectively but should be **`-`** for Sunday.
Input Arguments	**`file_name`** : [***str***] CSV file name of Work Sheet
Output Arguments	**`date_office_name_mapping`** : [***dict***] Mapping of the date and office name where he worked as { Key : Value } pair
Example Call	***date_office_name_mapping = readWorkSheet(csv_file_name)***

  - `calculateOfficeHrs()`

Function Name	**`calculateOfficeHrs()`**
Purpose	Calculate the number of hours worked in office A and B with the given mapping of date and office name.
Input Arguments	**`mapping_dict`** : [***dict***] Mapping of the date and office name where he worked as { Key : Value } pair
Output Arguments	**`(no_hrs_office_A, no_hrs_office_B)`** : [***tuple***] Number of hours worked in office A and B as pair
Example Call	***total_hrs_office_A_B = calculateOfficeHrs(date_office_name_mapping)***

  - `writeOfficeWorkSheet()`

Function Name	**`writeOfficeWorkSheet()`**
Purpose	Writes a CSV file with date and office name where the person worked on each day.
Input Arguments	**`mapping_dict`** : [***dict***] Mapping of the date and office name where he worked as { Key : Value } pair **`out_file_name`** : [***str***] File name of CSV file for writing the data to
Output Arguments	None, but writes the data of office name for each date in the CSV file with given name.
Example Call	***writeOfficeWorkSheet(date_office_name_mapping, out_csv_file_name)***

- To run and debug your solution, type the below command in Terminal:

  ```
  $ python3 assignment3.py
  ```

  This command will run the Python script `assignment3.py` .

- Refer the **Expected Output** section below and debug your code to get the correct output.

Expected Output

- The provided sample CSV file, `week4_assignment3_sample.csv` consists data with fields ***date***, ***office_name*** with ***office_name*** column as blank.

- The content of this CSV file is shown below:

  ```
  date,office_name
  2021-03-26,
  2021-04-01,
  2021-04-20,
  2021-04-04,
  2021-04-12,
  2021-04-23,
  2021-04-03,
  2021-03-29,
  2021-03-28,
  2021-03-31,
  2021-04-10,
  2021-04-16,
  2021-04-24,
  2021-04-11,
  2021-04-13,
  ```

- The expected output of program `assignment3.py` i.e., to write the CSV file with given name, defined in `main` function, having the fields ***date*** and ***office_name*** is shown below:

  ```
  {
      '2021-03-26': 'A', '2021-04-01': 'B', '2021-04-20': 'B', '2021-04-04': '-', '202
      '2021-04-23': 'A', '2021-04-03': 'B', '2021-03-29': 'A', '2021-03-28': '-', '202
      '2021-04-10': 'B', '2021-04-16': 'A', '2021-04-24': 'B', '2021-04-11': '-', '202
  }
  (48, 36)
  ```

- The first five lines above are the output of `readWorkSheet` function i.e. variable `date_office_name_mapping` and,

  the last line is the output of `calculateOfficeHrs` function i.e. variable `total_hrs_office_A_B` .

- The program also generates a CSV file with the name `output_week4_assignment3_sample.csv` as defined in `main` function that is then passed to `writeOfficeWorkSheet` function. The CSV file will have the contents as shown below:

  ```
  date,office_name
  2021-03-26,A
  2021-04-01,B
  2021-04-20,B
  2021-04-04,-
  2021-04-12,A
  2021-04-23,A
  2021-04-03,B
  ```

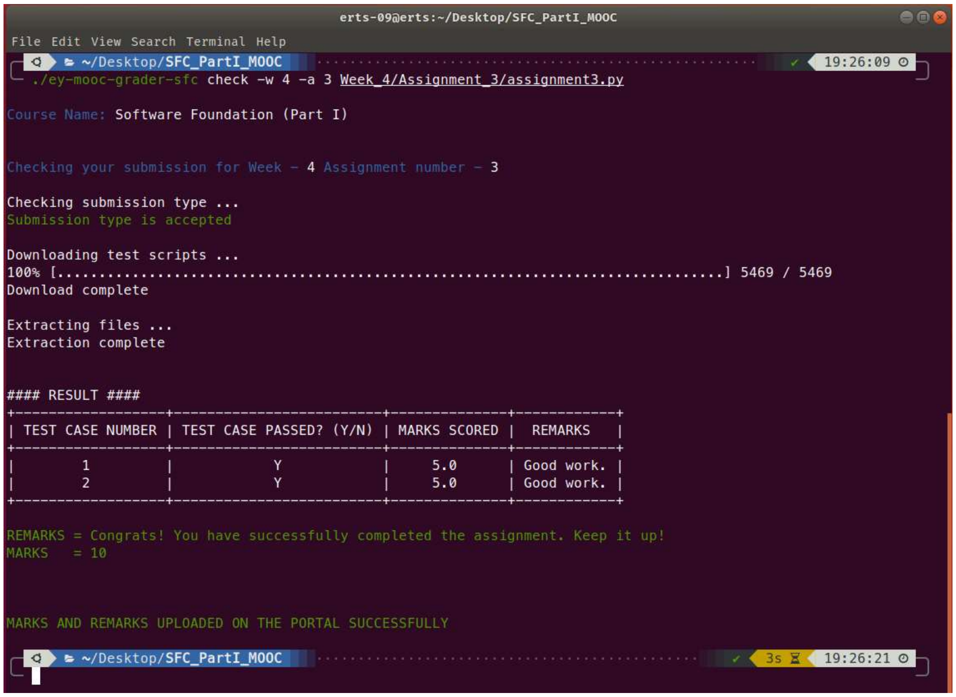
```
2021-03-29,A
2021-03-28,-
2021-03-31,A
2021-04-10,B
2021-04-16,A
2021-04-24,B
2021-04-11,-
2021-04-13,B
```

Grading and Submission Instructions

- Navigate to the folder where the ***ey-mooc-grader-sfc*** application resides.
- To grade your solution, run the **`check`** command of the application as follows:

```
$ ./ey-mooc-grader-sfc check -w 4 -a 3 Week_4/Assignment_3/assignment3.py
```

- This will run your program **`assignment3.py`** against random test cases and grade it. Marks and appropriate remarks will be provided as shown in Figure 1.
- Your program file **`assignment3.py`** , marks scored and remarks will get uploaded to the MOOC portal.

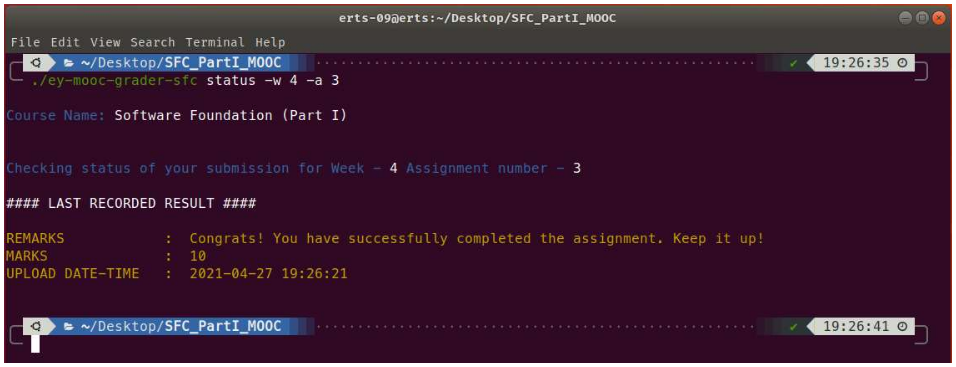

erts-09@erts:~/Desktop/SFC_PartI_MOOC
File Edit View Search Terminal Help

```
 ~/Desktop/SFC_PartI_MOOC ............................................. ✓ 19:26:09
 ./ey-mooc-grader-sfc check -w 4 -a 3 Week_4/Assignment_3/assignment3.py

Course Name: Software Foundation (Part I)


Checking your submission for Week - 4 Assignment number - 3

Checking submission type ...
Submission type is accepted

Downloading test scripts ...
100% [..........................................................................] 5469 / 5469
Download complete

Extracting files ...
Extraction complete


#### RESULT ####
+------------------+-------------------------+--------------+------------+
| TEST CASE NUMBER | TEST CASE PASSED? (Y/N) | MARKS SCORED |  REMARKS   |
+------------------+-------------------------+--------------+------------+
|        1         |            Y            |     5.0      | Good work. |
|        2         |            Y            |     5.0      | Good work. |
+------------------+-------------------------+--------------+------------+

REMARKS = Congrats! You have successfully completed the assignment. Keep it up!
MARKS   = 10



MARKS AND REMARKS UPLOADED ON THE PORTAL SUCCESSFULLY

 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 3s ⧗ 19:26:21
```

Figure 1: Output of running check command for Week 4 Assignment 3

- You can verify this by running the **`status`** command of the application as given below, refer Figure 2.

```
$ ./ey-mooc-grader-sfc status -w 4 -a 3
```



erts-09@erts:~/Desktop/SFC_PartI_MOOC
File Edit View Search Terminal Help

```
 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 19:26:35
 ./ey-mooc-grader-sfc status -w 4 -a 3

Course Name: Software Foundation (Part I)


Checking status of your submission for Week - 4 Assignment number - 3

#### LAST RECORDED RESULT ####

REMARKS            :  Congrats! You have successfully completed the assignment. Keep it up!
MARKS              :  10
UPLOAD DATE-TIME   :  2021-04-27 19:26:21


 ~/Desktop/SFC_PartI_MOOC ........................................ ✓ 19:26:41
```

Figure 2: Output of running status command for Week 4 Assignment 3

References

- Official Python documentation of CSV module
- Blog on Python CSV Module by RealPython
- Official Python documentation of Datetime module
- Blog on For Loops by RealPython
- Blog on Python Dictionary by Programiz
- Blog on Python Dictionary by RealPython
- Blog on Python Tuples by RealPython

All The Best!