Lab Assignment 11

Objective:

To work with searching and sorting in python

Assignment:

Write a python program to search for an employee from an input file based on SSN. Extend the program to read the employee id from the input file and sort and display the details. There are two input files – ***"Employee_SSN.txt"*** and ***"Employee_Details.txt"***.

Employee_SSN.txt should contain the following fields:

<Employee_ID>, <SSN_Number>, <Date_of_Birth>

Employee_Details.txt should contain the following fields:

<Employee_ID>, <Name>, <Date_of_Joining>, <Department>, <Annual_Salary>

All **<*Employee_ID*>** in ***"Employee_SSN.txt"*** should be there in ***"Employee_Details.txt"***.

Details of the program:

The program should display the following options to the user:

Option 1: Search for an employee using their **<*SSN_Number*>,** and display the employee details.

Option 2: Sort the **<*Employee_ID*>**, and display the employee details.s

Option 3: Exit

Based on each option the user selects, the following operations are to be performed.

1. When the user selects option 1:
    a. The program should prompt the user to enter an SSN number. The SSN number the user enters and stores in the file is a 9-digit number of the form (Sample SSN: 123-234-345).
    b. Check whether the entered SSN is in the file ***"Employee_SSN.txt"***.
    c. If there is a matching entry, then the details corresponding to the employee from ***"Employee_Details.txt"*** should be displayed, based on the **<*Employee_ID*>**.
    d. If there is no matching SSN found in the ***"Employee_SSN.txt"***, then *"SSN not found in the file"* is displayed.
2. When the user selects option 2:
    a. Open the file ***"Employee_Details.txt"***. Read the <Employee_ID> from the file and store it in a list. Sort the list and display the employee details corresponding to the sorted list.

b. Display the details from the file ***"Employee_Details.txt",*** based on sorted <***Employee_ID***> in the terminal.

3. When the user selects option 3, exit from the program.

Attention:

1. You can create your own input files or use the files uploaded on the blackboard.
2. Input file format is given below:

Sample "Employee_SSN.txt"

```
2001, 123-325-234, 06/07/1993
1001, 234-345-123, 08/12/1986
1003, 124-254-354, 11/27/1987
1006, 132-225-534, 06/17/1988
2008, 175-432-236, 05/04/1996
```

Sample "Employee_Details.txt"

```
2001, name1, 03/12/2017, Eng, 42000
1001, name2, 04/01/2016, Aero, 32000
2008, name3, 03/12/2018, Eng, 54000
1006, name4, 02/24/2016, Arts, 45000
1003, name5, 05/20/2014, Aero, 45000
```

Instructions:

- Preferred programming environment:
  - OS : Linux (Mint)
  - Interpreter : Python 3 (not Python 2)
  - Editor : gedit or editor of your choice
- The program is saved as a file with .py extension.
- The program should include a comment block at the top with your name, course number and course section, assignment number

  For example:

  ```
  # Your name
  # CSCI II 161 L01/L02
  # Assignment 11
  ```

- Upload your file as your ***lastname_firstname_assignmentnumber.py***

For example:

lastname_firstname_11.py

Sample output:

```
*****************************

1: Search an employee using SSN number
2: Sorting the employee based on Employee ID.
3: Exit

Please choose an option: 1
Enter the SSN #: 123-325-234

Great! Match found!


*****************


The Employee details are :
2001 -  name1 -  03/12/2017 -  CS -  42000
*****************************

1: Search an employee using SSN number
2: Sorting the employee based on Employee ID.
3: Exit

Please choose an option: 1
Enter the SSN #: 123-325-234

Great! Match found!


*****************


The Employee details are :
2001 -  name1 -  03/12/2017 -  CS -  42000

The Employee details are :
1001 -  name2 -  04/01/2016 -  EE -  32000
1006 -  name4 -  02/24/2016 -  AV -  45000
2001 -  name1 -  03/12/2017 -  CS -  42000
2008 -  name3 -  03/12/2018 -  CS -  54000
```