Assignment

Write a program to compute the employee management system. The requirements are listed below:

- This Employee Management System has 5 functions. Users are allowed to display all employee information, display company statistics, add and remove an employee, update employee salary by specify employee's ID
- To add an employee record, user is required to input employee's name, salary and department. Employee ID should be auto generate follow along with the last employee's record ID
- To remove an employee record, user is required to input the employee's ID
- To update an employee salary, user is required to input the employee's ID and salary.
- All employee information should be displayed for user as a reference when user perform remove employee and update employee's salary action
- The list of all employee information is stored in a text file named "employee_list.txt"

A sample output of normal flow is shown below:

```
==================== RESTART: C:/Python39/assignment.py ====================
Welcome to Employee Management System.
=======================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 1

All employee(s) information:
ID        | Name                | Salary      | Department
IVE00001 | Kelvin Yip           | $43,210.50  | IT
IVE00002 | Cow Leung            | $32,105.40  | Admin
IVE00003 | Leung Pig Hung       | $21,054.30  | HR
IVE00004 | Michael Fung         | $10,543.20  | Finance
IVE00005 | Joe Yeung            | $6,543.20   | IT
IVE00006 | Martin Kung          | $5,432.10   | Admin

=======================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 2
Please input employee's name, Enter to return: Pika Chiu
Please input employee's salary, Enter to return: 15000
Please input employee's department, Enter to return: IT
***** Employee Added Successfully *****
```

User's input

User's input

```
=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 3

All employee(s) information:
ID        | Name                | Salary      | Department
IVE00001 | Kelvin Yip           | $43,210.50  | IT
IVE00002 | Cow Leung            | $32,105.40  | Admin
IVE00003 | Leung Pig Hung       | $21,054.30  | HR
IVE00004 | Michael Fung         | $10,543.20  | Finance
IVE00005 | Joe Yeung            | $6,543.20   | IT
IVE00006 | Martin Kung          | $5,432.10   | Admin
IVE00007 | Pika Chiu            | $15,000.00  | IT

Please input employee's ID to remove record, Enter to return: IVE00005
***** Employee Removed Successfully *****

=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 4

All employee(s) information:
IVE00001 | Kelvin Yip           | $43,210.50  | IT
IVE00002 | Cow Leung            | $32,105.40  | Admin
IVE00003 | Leung Pig Hung       | $21,054.30  | HR
IVE00004 | Michael Fung         | $10,543.20  | Finance
IVE00006 | Martin Kung          | $5,432.10   | Admin
IVE00007 | Pika Chiu            | $15,000.00  | IT

Please input employee's ID, Enter to return: IVE00006
Please input employee's updated salary, Enter to return: 10000
***** Employee Salary updated Successfully *****

=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
```

User's input

Display all employee for user's reference

New employee added in previous step

User's input

User's input

Display all employee for user's reference

User's input

```
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 5

All employee(s) information:
ID       | Name                 | Salary      | Department
IVE00001 | Kelvin Yip           | $43,210.50  | IT
IVE00002 | Cow Leung            | $32,105.40  | Admin
IVE00003 | Leung Pig Hung       | $21,054.30  | HR
IVE00004 | Michael Fung         | $10,543.20  | Finance
IVE00006 | Martin Kung          | $10,000.00  | Admin
IVE00007 | Pika Chiu            | $15,000.00  | IT

Company statistics:
Department | No. of staff | Highest Salary | Lowest Salary | Average Salary
HR         |            1 | $      21054.3 | $     21054.3 | $      21054.3
IT         |            2 | $      43210.5 | $     15000.0 | $      29105.2
Admin      |            2 | $      32105.4 | $     10000.0 | $      21052.7
Finance    |            1 | $      10543.2 | $     10543.2 | $      10543.2

Total number of staff: 6
Highest Salary       : $43,210.5
Lowest Salary        : $10,000.0
Average Salary       : $21,985.6
=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit):
```

User's input

A sample output of exceptional case is shown below:

```
==================== RESTART: C:/Python39/assignment.py ====================
Welcome to Employee Management System.
=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 0
Invalid input for choice
Please input your choice. (1 - 5, Enter to exit): 2
Please input employee's name, Enter to return: Just 4 Fun
Not a valid employee name
```

Invalid Function Number

Name should not contain any digit

```
Employee's name should not contain digit
Please input employee's name, Enter to return: Pika Chiu
Please input employee's salary, Enter to return: 0
Employee's salary should be greater than 0
Please input employee's salary, Enter to return: 60000
Please input employee's department, Enter to return: Marketing
Not a valid department
Employee's department should be HR/IT/Admin/Finance
Please input employee's department, Enter to return: IT
***** Employee Added Successfully *****

=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit): 3

All employee(s) information:
ID        | Name                | Salary      | Department
IVE00001 | Kelvin Yip           | $43,210.50  | IT
IVE00002 | Cow Leung            | $32,105.40  | Admin
IVE00003 | Leung Pig Hung       | $21,054.30  | HR
IVE00004 | Michael Fung         | $10,543.20  | Finance
IVE00005 | Joe Yeung            | $6,543.20   | IT
IVE00006 | Martin Kung          | $5,432.10   | Admin
IVE00007 | Pika Chiu            | $60,000.00  | IT

Please input employee's ID to remove record, Enter to return: IVE00009
Not a valid employee's ID
Employee's ID record not found
Please input employee's ID, Enter to return:

=====================================
Employee Management System Menu:
No. | Function
1   | Display all employee
2   | Add an employee
3   | Remove an employee
4   | Update employee salary
5   | Display company statistics
Please input your choice. (1 - 5, Enter to exit):
```

Salary should be greater than $0

Department restricted to HR/IT/Admin/Finance

Employee does not exist

Follow the steps below to implement the program:

1. Declare a **constant tuple** named **DEPARTMENT** to store the department that can be selected when add an employee record. Department includes **HR**, **IT**, **Admin** and **Finance**.

2. Declare the following constants for index of functions in the main menu.
   RETURN = ""
   DISPLAY_ALL_EMPLOYEES = 1
   ADD_AN_EMPLOYEE = 2
   REMOVE_AN_EMPLOYEE = 3
   UPDATE_EMPLOYEE_SALARY = 4
   DISPLAY_STATISTICS = 5

3. Declare a **list of dictionary** named **list_dict_employees** in main function to store all employee information. You are NOT allowed to make the list_dict_employees variable global. The following diagrams show the detailed structure of **DEPARTMENT** and **list_dict_employees**:

tuple DEPARTMENT

0	"HR"
1	"IT"
2	"Admin"
3	"Finance"

list of dictionary **list_dict_employees**

0 — employee 1 record

Key	Data Type	Sample Data
ID	String	"IVE00001"
Name	String	"Kelvin Yip"
Salary	Float	43210.5
Department	String	"IT"

1 — employee 2 record

Key	Data Type	Sample Data
ID	String	"IVE00002"
Name	String	"Cow Leung"
Salary	Float	32105.4
Department	String	"Admin"

…

n — employee n record

Key	Data Type	Sample Data
ID	String	"IVE000xx"
Name	String	"xxxxx xxxxx"
Salary	Float	xxxxx.xx
Department	String	"xxxxx"

4. Write a function **display_all_employees** to display all employee records for the menu item 1 – **Display all employee** function as shown in the above sample output. This function should also be called before remove an employee, update employee salary and display company statistics. The **list of employees (i.e. list_dict_employee)** should also be passed into this function when calling.

5. Write a function **read_employee_from_file** to read in employee's information from file named "employee.txt" located in the same folder. The sample file "employee.txt" is given for you already and should follow the same format. **Zero marks** will be given if you assigned the employee data by static variable such as **list_dict_employees = {…}**

6. The main function should contain the following logics:
   - Read in employee's information from file named "employee.txt" located in the same folder.
   - Display a welcome message and followed by the main menu of employee management system. Users can display all employee's information, add and remove employee, update employee salary and display company statistics.
   - To add an employee, user is required to input employee's name, salary and department. **Employee ID should be auto-generated follow along with the last employee's employee ID**
   - To remove an employee record, user is required to input the employee's ID.
   - To update an employee salary, user is required to input the employee's ID and salary.
   - Display all employee and company's employee statistics on request. For details please reference to sample output provided.
   - Handle all errors inputted by user and **ask user to correct and input the same data again immediately**, such as but not limited to incorrect menu function number, incorrect employee name, incorrect employee ID, incorrect salary, incorrect department, etc.

7. Complete the corresponding parts in the main() function to perform the add employee function for menu item 2 – **Add an employee**. This part does at least the followings:
   - Ask user to input for employee's name, employee's salary and employee's department. Return to previous menu when user pressed Enter directly and validate user's input where necessary.
   - Obtain the employee ID from the last employee ID in the system and calculate the employee ID of the new employee ID
   - Create the employee record based on user's input and new employee ID calculated (should be in dictionary data type)
   - Append the new created employee record to the **list of employees**

8. Complete the corresponding parts in the main() function to perform the remove employee function for menu item 3 – **Remove an employee**. This part does at least the followings:
   - Display all employees by calling the **display_all_employees** function for user to choose the employee to remove.
   - Ask user to input for employee's ID. Return to previous menu when user pressed Enter directly and validate user's input where necessary.
   - Loop through the list of employees and remove the employee record from the list if exist.

9. Complete the corresponding parts in the main() function to perform the update employee salary function for menu item 4 – **Update employee salary**. This part does at least the followings:
   - Display all employees by calling the **display_all_employees** function for user to choose the employee to perform salary update.
   - Ask user to input for employee's ID and employee's updated salary. Return to previous menu when user pressed Enter directly and validate user's input where necessary.
   - Loop through the list of employees and update the employee's salary in the list if exist.

10. Complete the corresponding parts in the main() function to perform the display company statistics function for menu item 5 – **Display company statistics**. This part does at least the followings:
    - Display all employees information by calling the **display_all_employees** function
    - Display the company statistics by showing the no. of staff, highest salary, lowest salary and average salary **in each department**.
    - Display the total number of staff, highest salary, lowest salary and average salary in this company.

Important Hints:

You may **implement some minor functions at last**, such as **reading employee's information from the file**, and most of the **input validation**. Such as validation of new add employee name, salary, department, employee ID for remove and update function, etc.

Instruction to students:

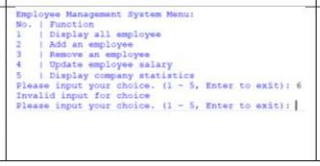
1. This is an End of Module Assessment and the weighting of this assignment is 20% of the Module Mark.

2. This assignment should be done by each individual student. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score Zero marks.

3. You must use Python 3 to develop the programs.

4. Your programs must follow the style guide stated in PEP8 – Style Guide for Python Code published by python.org. https://www.python.org/dev/peps/pep-0008/. Marks may be deducted if the style guide is not followed.

5. You are required to hand in:
   - Well-commented source code.
   - A test plan showing the evidence of testing.

ID	Test Case Name	Procedure	Expected Output	Screen Dump	Result
1	Display all employees in the system	1. In the employee management system menu, input '1' for display all employee	All employees' information including their employee ID, name, salary and department should be displayed		Pass / ~~Fail~~
2	Add an employee with name "Pika Chiu", salary 15000 and works in IT department into the system.	1. In the employee management system menu, input '3' to add an employee 2. Input 'Pika Chiu' for employee's name 3. Input "15000" for salary 3. Input "IT" for IT department	A success message "Employee Added Successfully" should be displayed		Pass / ~~Fail~~
	…	…	…		
5	Input invalid function number in main menu	1. In the employee management main menu, input '5'	An error message "Invalid input for choice" should be displayed and will ask the user to input function number again.		Pass / ~~Fail~~

6. Prepare a word document with a number of test cases showing different inputs for different situations that your program may encounter and how your program responses to show the capability of your program.

7. For each test case, states the objective of the test case, input data and expected result. You should also include screen dump for each test run as evidence.

8. Submit all your works (in a zip file under the name of your student ID – e.g. 229999999.zip) to the Moodle website (http://moodle2223.vtc.edu.hk) by 11:59pm, 22 November 2022 (Tuesday).

9. Each student will be required to conduct an assignment demonstration during laboratory class to show the system and walk through all the functions. The date of the demonstration will be after the assignment submission and is to be confirmed later (It should be the next lab session just after the submission of assignment normally). Zero Marks will be given if students do not perform any assignment demonstration.

10. Marks Distribution
    - System Implementation (70%)
    - Validation on the input data and display appropriate error messages (20%)
    - Test plan (10%)