L10 Tasks

L10-T1: Employee dictionary

L10-T2: Employee dictionary 2

L10-T3: Titanic Passenger Data Processing

L10-T4: JSON file

L10-T5: Creating your own modules

Submission:

- Submit **all** the exercises to CodeGrade via Moodle before the deadline.

Note:

- If you see some "Code Structure Tests" hidden in CodeGrade but they are not mentioned in the task description, you don't need to worry about them. They are just there to make sure the code is ok.

- Be especially careful with spaces so that your output follows the sample output. Note that also each input-string in the program is ending with '\n'. The reason for this is that it makes the output in CodeGrade more readable

L10-T1: Employee dictionary

Write a function `employee_dictionary(dict_list)`, which asks the user the name of the worker, his workplace, and his age. The function creates from these values a dictionary and appends it to the `dict_list`. Note that `dict_list` is a (mutable) list, so you **do not** need to return it.

Write a main program calling this function. In the beginning, the list of workers is empty. The program should also ask first how many employees the user wants to add.

Write also a function `print_work_info(dict_list)`, which prints the information about all the workers according to the example below.

Note: “Code structure tests” will be used to make sure that the program has all the required functions, so please make sure that your program has the right functions with exact names and parameters.

Example run 1:

```
How many employees do you want to add?:
4
Enter worker's name:
John
Enter worker's workplace:
LUT
Enter worker's age:
40
Enter worker's name:
Olympic Jelly
Enter worker's workplace:
Paris
Enter worker's age:
56
Enter worker's name:
Grace Lily
Enter worker's workplace:
Hel Airplane
Enter worker's age:
79
Enter worker's name:
Duck Term
Enter worker's workplace:
JBL
Enter worker's age:
70
List of Employees:
Name: John, Workplace: LUT, Age: 40
Name: Olympic Jelly, Workplace: Paris, Age: 56
Name: Grace Lily, Workplace: Hel Airplane, Age: 79
Name: Duck Term, Workplace: JBL, Age: 70
```

L10-T2: Employee dictionary 2

In L10-T1, you have learned how to create and add new data to the `employee_list`. In this task, we will try to process this `employee_list` a little bit. Here is the `employee_list` that you have to use in your code:

```
employee_dict =
[
{'Name': 'John', 'Workplace': 'LUT', 'Age': 25},
{'Name': 'Jack', 'Workplace': 'Finnair', 'Age': 18},
{'Name': 'Robin', 'Workplace': 'JBL', 'Age': 32},
{'Name': 'Annie', 'Workplace': 'LUT', 'Age': 24},
{'Name': 'Niels', 'Workplace': 'Microsoft', 'Age': 45}
]
```

Write a menu-based program that has the following functions:

1) `remove_employee(employee_list)`: Remove a selected employee from the list. The program should first provide the user with a list of employees with their position numbers in the list, then ask the user to choose which employee needs to be removed from the list. Error handling should be used to catch any possible error.

2) `modify_employee(employee_list)`: Modify employee's data. The program should also provide the user with a list of employees with their position numbers in the list, and then ask the user to choose which employee's data needs to be modified. After that, it asks the user to choose which information between "Workplace" or "Age" that will be modified. Then, the user can enter new data which will be written to the list. Error handling should be used to catch any possible error.

3) `print_employee(employee_list)`: Prints the information about all the workers according to the example below.

Note: "Code structure tests" will be used to make sure that the functions are created and used in the right way. Make sure that your functions have exact names and parameters. If those tests are not passed, the other IO tests will not be graded.

Example run 1 (remove employee):

```
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
3
List of Employees:
```

```
Name: John, Workplace: LUT, Age: 25
Name: Jack, Workplace: Finnair, Age: 18
Name: Robin, Workplace: JBL, Age: 32
Name: Annie, Workplace: LUT, Age: 24
Name: Niels, Workplace: Microsoft, Age: 45
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
1
List of Employees:
1) Name: John, Workplace: LUT, Age: 25
2) Name: Jack, Workplace: Finnair, Age: 18
3) Name: Robin, Workplace: JBL, Age: 32
4) Name: Annie, Workplace: LUT, Age: 24
5) Name: Niels, Workplace: Microsoft, Age: 45
Enter the number of the employee to remove:
2
Removed employee: Jack
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
2
List of Employees:
1) Name: John, Workplace: LUT, Age: 25
2) Name: Robin, Workplace: JBL, Age: 32
3) Name: Annie, Workplace: LUT, Age: 24
4) Name: Niels, Workplace: Microsoft, Age: 45
Enter the number of the employee to modify:
4
Enter the field to modify:
1) Workplace
2) Age
1
Enter new value for Workplace:
Helsinki
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
```

```
3
List of Employees:
Name: John, Workplace: LUT, Age: 25
Name: Robin, Workplace: JBL, Age: 32
Name: Annie, Workplace: LUT, Age: 24
Name: Niels, Workplace: Helsinki, Age: 45
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
0
See you again!
```

Example run 2 (error handling):

```
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
1
List of Employees:
1) Name: John, Workplace: LUT, Age: 25
2) Name: Jack, Workplace: Finnair, Age: 18
3) Name: Robin, Workplace: JBL, Age: 32
4) Name: Annie, Workplace: LUT, Age: 24
5) Name: Niels, Workplace: Microsoft, Age: 45
Enter the number of the employee to remove:
something
Invalid input. Please enter a number.
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
2
List of Employees:
1) Name: John, Workplace: LUT, Age: 25
2) Name: Jack, Workplace: Finnair, Age: 18
3) Name: Robin, Workplace: JBL, Age: 32
4) Name: Annie, Workplace: LUT, Age: 24
5) Name: Niels, Workplace: Microsoft, Age: 45
Enter the number of the employee to modify:
```

```
not_something
Invalid input. Please enter a number.
Menu:
1) Remove an employee
2) Modify employee data
3) Print all employees
0) Exit
Enter your choice:
0
See you again!
```

L10-T3: Titanic Passenger Data Processing

Your task is to deal with the data of Titanic passengers. We have an existing CSV file containing the details of all passengers aboard the infamous Titanic cruise on 15 April 1912. You can read more about Titanic from Wikipedia: https://en.wikipedia.org/wiki/Titanic.

The file `titanic.csv` is available fully only in CodeGrade, so please don't submit any CSV files to CodeGrade. In Moodle, you can find the file `titanic_sample.csv`, which contains the first 15 rows of the file. Your task is to find:

(a) number of male passengers

(b) the number of female passengers

(c) the average age of all passengers (rounded to the closest integer)

(d) the age of the oldest passenger (as an integer).

When you download the file `titanic_sample.csv` from Moodle to your computer, please rename it to `titanic.csv`, and then use this file's name in your program. By that, when you submit your solution to CodeGrade, it can run your file normally without any trouble.

Note that the age of some passengers is missing. In such a case, you just ignore these rows.

Note: "Code structure tests" will be used to make sure that the program has all the required requirements.

Example run 1 (titanic_sample.csv):

```
The number of male passengers: 5
The number of female passengers: 5
The average age of passengers: 28
The age of the oldest passenger: 54
```

L10-T4: JSON file

The file `students.json` is only available in CodeGrade. It contains JSON items looking like below:

```
{
    "id": 13,
    "name": "Ethan Davis",
    "age": 18,
    "grade": "B",
    "courses": ["Math", "Physics", "Chemistry"]
}
```


These JSON items are packed inside a list. You can find a shorter version of this file in Moodle, its name is `students_short.json`.

Same as L10-T3, please rename the file `students_short.json` to `students.json` after downloading it from Moodle.

Write a Python program that imports this JSON file (`students.json`) and create some functions following these requirements:

1. `print_students_by_age(student_list)`: this function will receive the student list as the parameter, then it asks the user to choose the age of the students and print data of those having that age.

2. `print_students_by_course(student_list)`: this function will receive the student list as the parameter, then it asks the user to choose a course that students enrolled in, and it then prints their data.

3. `print_students_name(student_list)`: this function will receive the student list as the parameter and print the data of those whose first name ends with the letter "a". The names of the students are given in the form: `firstname lastname`.

Note: "Code structure tests" will be used to make sure that the functions are created and used in the right way. Make sure that your functions have exact names and parameters. If those tests are not passed, the other IO tests will not be graded.

Note: The example below contains only a partial answer. The red dashed lines denote that lines are removed to save space.

Example run 1:

```
What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
1
Select the ages of the students:
```

```
1) 19
2) 20
3) 21
4) 22
Enter your selection:
1
Student ID: 2, Name: Jane Doe, Age: 19
Student ID: 7, Name: Daniel Lee, Age: 19
- - -
Student ID: 58, Name: Chloe Brown, Age: 19
Student ID: 62, Name: Aria Smith, Age: 19

What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
1
Select the ages of the students:
1) 19
2) 20
3) 21
4) 22
Enter your selection:
3
Student ID: 3, Name: David Johnson, Age: 21
Student ID: 9, Name: James Wilson, Age: 21
- - -
Student ID: 59, Name: Ethan Lee, Age: 21
Student ID: 63, Name: Daniel Anderson, Age: 21

What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
2
Select the course:
1) Computer Science
2) History
3) Math
4) Art
Enter your selection:
2
```

```
Student ID: 2, Name: Jane Doe, Course: ['History', 'English', 'Biology']
Student ID: 7, Name: Daniel Lee, Course: ['History', 'English', 'Biology']
- - -
Student ID: 59, Name: Ethan Lee, Course: ['History', 'English', 'Biology']
Student ID: 63, Name: Daniel Anderson, Course: ['History', 'English', 'Biology']

What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
2
Select the course:
1) Computer Science
2) History
3) Math
4) Art
Enter your selection:
1
Student ID: 3, Name: David Johnson, Course: ['Computer Science', 'Statistics', 'Economics']
Student ID: 5, Name: Michael Brown, Course: ['Physics', 'Chemistry', 'Computer Science']
- - -
Student ID: 58, Name: Chloe Brown, Course: ['Computer Science', 'Statistics', 'Economics']
Student ID: 62, Name: Aria Smith, Course: ['Computer Science', 'Statistics', 'Economics']

What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
3
Students whose name end with 'a':
Student ID: 6, Name: Sophia Anderson
Student ID: 8, Name: Olivia Davis
- - -
Student ID: 54, Name: Ava Wilson
Student ID: 62, Name: Aria Smith
```

```
What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
5
Invalid choice. Please try again.

What do you want to do?
1) Print students by age
2) Print students based on the course they are taking
3) Print students whose first names ends with the letter "a"
0) Stop the program
Enter your choice:
0
See you again!
```

L10-T5: Creating your own modules

In this task, you need to prepare and submit two files and please **make sure** that you use these names. The file **`person.py`** is a module that contains the definition of the class Person. The objects of this class have the ***attributes*** name and age. The class has three ***methods*** which are:

- `__init__(self, name, age)`: which creates a new person object.
- `introduce(self)`: which prints the person's name and age to the screen as demonstrated in the example run below.
- `celebrate_birthday(self)`: which increases the age of this person by one.

Then write a program **`L10-T5.py`**, which first imports the module person. Then you need to create to two persons whose names are Valtteri and Kimi. Their ages are 35 and 45, respectively. Then you let these persons to introduce themselves. Because Kimi's birthday is 17th October, you call the celebrate_birthday() method with the person whose name is Kimi and let this person introduce himself again.

Note: “Code structure tests” will be used to make sure that the methods are created and used in the right way. Make sure that your methods have exact names. If those tests are not passed, the other IO tests will not be graded.

Example run 1:

```
My name is Valtteri, and I am 35 years old.
My name is Kimi, and I am 45 years old.
My name is Kimi, and I am 46 years old.
```