Exercises

Week 40-41

DM536 Introduction to Programming
DM562 Scientific Programming
DM857 Introduction to Programming
DS830 Introduction to Programming

1 Phone Book Manager

In this exercise you will develop a program for managing a phone book (numbers follow the 8-digit format with optional spaces separating pairs of digits (e.g.,`'12345678'`, `'1234 5678'`, `'12 34 56 78'`). The program consists of two modules:

- `phone_book`, is responsible for maintaining the phone book in memory and to do so it offers functions for searching, adding, and updating entries of the phone book (documented below).
- `manager`, is the main module of program and is responsible for handling the interaction with the user (you will implement this module, instructions are at the end).

1.1 Module `phone_book`

The module `phone_book` provides the following functions.

- `get(name)` returns the phone number stored under the given name provided it is present. The function raises `ContactNotFoundError` if there is not an entry for `name`.
- `add(name,phone)` adds a new entry to the phone book provided there is no entry for the given name. The function raises `DuplicateContactError` if there is already an entry for `name` and `ValueError` if the string `phone` is not a phone number (8-digit format).
- `update(name,phone)` updates an existing entry. The function raises `ContactNotFoundError` if there is no entry for `name` and `ValueError` if the string `phone` is not a phone number (8-digit format).
- `is_phone_number(phone)` Checks if the given string is an 8-digit phone number.

The module defines the following error types.

- `ContactNotFoundError`
- `DuplicateContactError`

1.2 Module `manager`

Your task will be to write `manager` (using `phone_book`) and following a bottom-up approach.

1. Define a function `search()` that asks the user for a name to search in the phone book and prints the corresponding number or a message if there is no entry for the given name.

2. Define a function `input_phone_number(message)` that queries the user for a phone number adn returns it. The argument `message` contains the text to be displayed to the user to when asking for input.

3. Define a function `add()` that asks the user for a name and a phone number and then attempts at creating a new entry in the phone book.

4. Define a function `edit()` that asks the user for a name and a phone number and then updates the corresponding entry in the phone book, if there is any.

5. Define a function `add_or_edit()` that asks the user for a name and a phone number and then creates a new entry or updates an existing one.

6. Define a function `main()` that implement a prompt where the user can issue the following commands:

   - `search` to search the phone book by name (use your function `search`).
   - `add` to create a new entry (use your function `add`).
   - `edit` to edit an existing entry (use your function `edit`).
   - `set` to create or edit an entry regardless of whether it is already in the phone book (use your function `add_or_edit`).
   - `quit` to terminate the program (see function `exit` of module `sys`).

   After executing the corresponding actions, the prompt must be ready to accept a new command from the user (except after `quit`).

7. Complete the module with instructions for calling `main` if the module is run as a program.