CSC1015F Assignment 11

Consolidation (Functions, Lists, Searching and Sorting, Files)

Assignment Instructions

This assignment involves constructing Python programs that use basic data structures such lists, searching and sorting as well as file IO to manipulate data.

The assignment has one question that involves the use of functions to implement the following data structures; lists, searching and sorting, as well the file IO techniques as we have seen in Assignment 10. Some of your methods will specifically be tested individually by the automarker so make sure that you follow instructions carefully in each case.

NOTE Your solutions to this assignment will be evaluated for correctness and for the following qualities:

- Documentation
  - Use of comments at the top of your code to identify program purpose, author and date.
  - Use of comments within your code to explain each non-obvious functional unit of code.
- General style/readability
  - The use of meaningful names for variables and functions.
- Algorithmic qualities
  - Efficiency, simplicity

These criteria will be manually assessed by a tutor and commented upon. In this assignment, up to 10 marks will be deducted for deficiencies.

Today, the most common way of keeping in touch with other colleagues, family and friends is through digital communications. Many people use the contacts app on their phones, tablets or computers. In this assignment, we will be implementing a simple contact manager to store, process and retrieve a person's names, contact number, and email address. This information will be stored in a file, and we will be employing the concepts we have learned from file input and output in this case.

NOTE: The automatic marker will test each of your functions individually. To enable this, you MUST HAVE the following lines at the end of your program:

```
if __name__=='__main__':
    main()
```

Your Task:

Write a program called '`contacts.py`' by implementing the following method.:

- `path_exists(file_path)`

  The main purpose of this function is file handling. It checks if the path to the contacts file exists. It returns True of the path is found and false otherwise. For example, if we test this function using Python Shell on Wing 101 IDE, we get the following:

```
>>> from contacts import path_exists
>>> path_exists("contacts")
False
>>> path_exists("contacts.txt")
True
```

  The function returns false if the contacts file has not been created or the name has been incorrectly specified.

- `add_contact(file_path, name, phone, email)`

  This function allows the use to add a new contact to the file. For example, we can add a new contact as follows, using the Python Shell on Wing 101 IDE:

```
>>> from contacts import add_contact
>>> add_contact("contacts.txt", "Monaheng Lebeko",
    "0830930905", "monaheng@gmail.com")
    Contact added successfully.
```

- `custom_sort(contacts)`

  This function sorts the contacts list alphabetically by name. It returns as list of sorted contacts. For example, if we query the contacts.txt file from the Python Shell on Wing 101 IDE, we get the following:

```
>>> from contacts import custom_sort
>>> custom_sort(["Lebeko", "Keegan", "Monaheng", "Lireko"])
    ['Keegan', 'Lebeko', 'Lireko', 'Monaheng']
```

  Note that the list returned is sorted in alphabetical order.

- `search_contact(file_path, query)`

  This functions searches for contacts in the contacts file based on the query supplied by the user. In this task, we seek to implement a ternary search to find contacts matching the query. This is a variation of binary search in which the list is divided into three equal parts instead of two. Ternary search is more effective on sorted data.

- `read_contacts(file_path)`

  This functions reads contacts from the contacts file. The function returns a list of contacts read from the file.

- `list_contacts(file_path)`
  This function lists all contacts sorted alphabetically by name. From the Python Shell on Wing 101 IDE, we have:

```
>>> from contacts import list_contacts
>>> list_contacts("contacts.txt")

List of contacts:
============================================================
| Name                  | Phone           | Email                    |
============================================================
| Keegan Joubert       | 0924539857      | jbrkee001@myuct.ac.za     |
------------------------------------------------------------
| Lebeko Poulo | 0782554182      | lebeko.poulo@gmail.com    |
------------------------------------------------------------
| Lireko Majara        |  0982345768     |  lmajara@gmail.com        |
------------------------------------------------------------
| Lisebo Thamae        | 00266578916194  | lthamae@gmail.com         |
------------------------------------------------------------
| Monaheng Lebeko      | 0830930905      | monaheng@gmail.com        |
------------------------------------------------------------
| Neo Henry            | 0672136547      | neohenry@gmail.com        |
------------------------------------------------------------
| Stephan Jamieson     | 0843256789      | sjamieson@cs.uct.ac.za    |
------------------------------------------------------------
| Vincent Mahlare      | 0892345673      | vmahlare@gmail.com        |
------------------------------------------------------------
```

- `wildcard function`
  This function matches a query string from a list with a wildcard character against a text string. It returns True of the query matches the text, False otherwise. This function was implemented in Assignment 5: Strings. You can adapt your program from Assignment 5 to implement this function.

You are welcome to implement any additional functions that serve to simplify your solution. Remember, it is best to follow the divide and conquer approach in solving problems. The smaller the problem, the easier it is to solve, and at the end, you put together the smaller pieces to make up the final solution to your problem.

The expected behaviour of the overall program is as shown below. As an example, the file contacts.txt is used to store or save contact details for several individuals.

Sample IO *(The input from the user is shown in* ***bold*** *font – do not program this*):

```
Enter the name for the contacts file:
contacts

1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 1
Enter name: Aslam Safla
Enter phone number: 0216503344
Enter email: asafla@cs.uct.ac.za
Contact added successfully.

1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 2
Enter first name, last name, phone number, or email to search:
Jamieson

Found contact(s):
============================================================
| Name                | Phone           | Email                    |
============================================================
| Stephan Jamieson    | 0843256789      | sjamieson@cs.uct.ac.za   |
------------------------------------------------------------

1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 3

List of contacts:
============================================================
| Name                | Phone           | Email                    |
============================================================
| Aslam Safla         | 0216503344      | asafla@cs.uct.ac.za      |
------------------------------------------------------------
| Keegan Joubert      | 0924539857      | jbrkee001@myuct.ac.za    |
------------------------------------------------------------
| Lebeko Poulo | 0782554182      | lebeko.poulo@gmail.com    |
------------------------------------------------------------
| Lireko Majara       |  0982345768     |  lmajara@gmail.com       |
------------------------------------------------------------
| Lisebo Thamae       | 00266578916194  | lthamae@gmail.com        |
------------------------------------------------------------
| Monaheng Lebeko     | 0830930905      | monaheng@gmail.com       |
------------------------------------------------------------
| Neo Henry           | 0672136547      | neohenry@gmail.com       |
------------------------------------------------------------
| Stephan Jamieson    | 0843256789      | sjamieson@cs.uct.ac.za   |
------------------------------------------------------------
```

```
| Vincent Mahlare      | 0892345673      | vmahlare@gmail.com        |
-------------------------------------------------------------

1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 4
Exiting program.
```

Sample IO *(The input from the user is shown in* ***bold*** *font – do not program this*):

```
Enter the name for the contacts file:
contacts


1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 2
Enter first name, last name, phone number, or email to search:
Hussein
No contact found.


1. Add Contact
2. Search Contact
3. List Contacts
4. Exit
Enter your choice: 4
Exiting program.
```

Submission

Create and submit a Zip file called '`ABCXYZ123.zip`' (where ABCXYZ123 is YOUR student number) containing `contacts.py`.