

## Program Instructions

1. Write a Python program that performs as an Employee Contact List which contains a list of contacts that can be modified or deleted.
2. Create a `contacts` module to meet the following requirements:
  - i. Create a file named `contacts.py`.
  - ii. Add a comment at the top of the file which indicates your name, date and the purpose of the file.
  - iii. Note: All contact lists within this module should assume the list is of the form: `[["first name", "last name"], ["first name", "last name"], ...]`
  - iv. Note: All functions should implement a docstring with a simple sentence that describes the function.
  - v. Define a function named `add_contact` to meet the following requirements:
    - a. Take a contact list as a positional parameter.
    - b. Take a `first_name` as a keyword parameter.
    - c. Take a `last_name` as a keyword parameter.
    - d. Append the `first_name/last_name` contact to the contact list.
  - vi. Define a function named `modify_contact` to meet the following requirements:
    - a. Take a contact list as a positional parameter.
    - b. Take a `first_name` as a keyword parameter.
    - c. Take a `last_name` as a keyword parameter.
    - d. Take an `index` as a keyword parameter.
    - e. If the index it is within the range of the contact list, modify the appropriate index of the contact list with the `first_name/last_name` contact, and return a `True`.
    - f. If the index it is not within the range of the contact list, return a `False` without modifying the contact list.
  - vii. Define a function named `delete_contact` to meet the following requirements:
    - a. Take a contact list as a positional parameter.
    - b. Take an `index` as a keyword parameter.
    - c. If the index it is within the range of the contact list, delete the contact at the index value, and return a `True`.
    - d. If the index it is not within the range of the contact list, return a `False` without modifying the contact list.

- viii. Define a function named `sort_contacts` to meet the following requirements:
  - a. Take a contact list as a positional parameter.
  - b. Take a `column` as a keyword parameter.
  - c. If the column is 0, sort the contact list by the first name.
  - d. If the column is 1, sort the contact list by the last name.
- 3. Create a main driver program to meet the following requirements:
  - i. Create a file named `main.py`.
  - ii. Add a comment at the top of the file which indicates your name, date and the purpose of the file.
  - iii. Import the `contacts` module.
  - iv. Define a variable to use for the contact list.
  - v. Implement a menu within a loop with following choices:
    - a. Print list
    - b. Add contact
    - c. Modify contact
    - d. Delete contact
    - e. Sort list by first name
    - f. Sort list by last name
    - g. Exit the program
  - vi. Prompt the user for the menu choice.
  - vii. Prompt the user for the information needed for the appropriate `contacts` function and call the function.
  - viii. Print out appropriate errors with function return values of `False`.
- 4. Run the program and repeat the steps above until you are satisfied your program output meets the above requirements.

Typical input and output for the program:

```
*** EMPLOYEE CONTACT MAIN MENU
```

- 1. Print list
- 2. Add contact
- 3. Modify contact
- 4. Delete contact
- 5. Sort list by first name
- 6. Sort list by last name
- 7. Exit the program

Enter menu choice: 2

Enter first name: Richard  
Enter last name: Stallman

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 2

Enter first name: Bill  
Enter last name: Gates

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 2

Enter first name: Steve  
Enter last name: Jobs

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 1

===== CONTACT LIST =====

Index	First Name	Last Name
0	Richard	Stallman
1	Bill	Gates
2	Steve	Jobs

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list

2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 3

Enter the index number: 2

Enter first name: Tim

Enter last name: Cook

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 1

```
===== CONTACT LIST =====
Index   First Name      Last Name
=====
0       Richard      Stallman
1       Bill         Gates
2       Tim          Cook
```

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 3

Enter the index number: 5

Enter first name: Mickey

Enter last name: Mouse

Invalid index number.

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact

5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 6

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 1

===== CONTACT LIST =====

Index	First Name	Last Name
0	Tim	Cook
1	Bill	Gates
2	Richard	Stallman

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 5

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 1

===== CONTACT LIST =====

Index	First Name	Last Name
0	Bill	Gates
1	Richard	Stallman
2	Tim	Cook

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 4

Enter the index number: 1

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 1

```
===== CONTACT LIST =====
Index   First Name           Last Name
=====
0       Bill                Gates
1       Tim                 Cook
```

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name
6. Sort list by last name
7. Exit the program

Enter menu choice: 4

Enter the index number: 5  
Invalid index number.

\*\*\* EMPLOYEE CONTACT MAIN MENU

1. Print list
2. Add contact
3. Modify contact
4. Delete contact
5. Sort list by first name

6. Sort list by last name
7. Exit the program

Enter menu choice: 7

## Submission

When completed, upload the two files into Canvas:

- main.py
- contacts.py