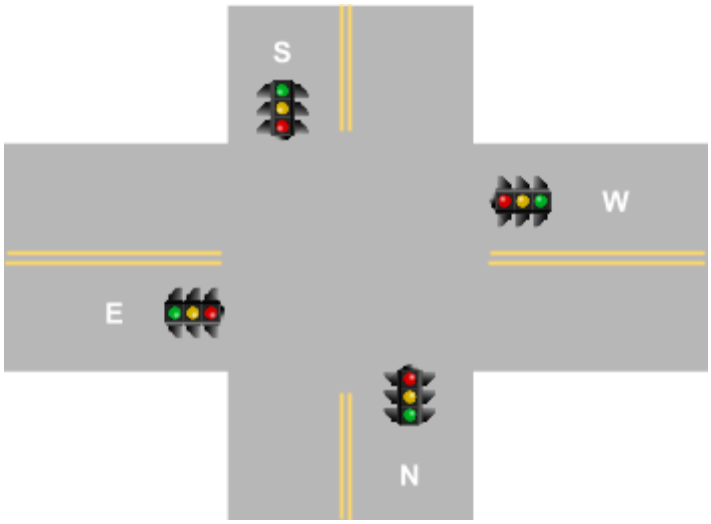
MATH5061: Week 4 Assignment

Instructions

Please submit answers to this assignment as a .tar.gz file of the directory containing the python solution source files via email to `math5061@temple.edu`. Make sure to use the subject line (without quotes) **"MATH5061:Assignment 04:ACCESSID"**
Where ACCESSID is your AccessNet ID, for example tue86537

Q1

Create a Python class `TrafficLight` that simulates the sequence of traffic signals at the intersection of 2 two-way streets. The class should accept one argument during instantiation called '`mode`' that can take either a 'US' or 'UK' string value with 'US' being the default if no mode argument is specified. Implement a method called `next()`, which returns the next state of all the lights as a dictionary with keys 'N', 'S', 'E' and 'W' which can take any of the 4 values 'Red', 'Amber', 'Green', or 'Red-Amber'.



In US mode the traffic lights should loop through the sequence below:

```
N             S             E             W
-----------------------------------------------
Green         Green         Red           Red
Amber         Amber         Red           Red
Red           Red           Red           Red
Red           Red           Green         Green
Red           Red           Amber         Amber
Red           Red           Red           Red
```

In UK mode the traffic light should loop through the sequence below:

```
N             S             E             W
-----------------------------------------------
Green         Green         Red           Red
Amber         Amber         Red           Red
Red           Red           Red-Amber     Red-Amber
Red           Red           Green         Green
Red           Red           Amber         Amber
Red-Amber     Red-Amber     Red           Red
```

Q2

Create a `ContactsDB` class that maintains a list of people. Each person is represented by a name. Each person has a dictionary type associated with them that can contain keys like ‘address’, ‘phone number’, ‘employer’ etc. The class should have the following methods and behaviour defined

- `add_person(pname, attr={})`
  Creates a new entry in the database with the name pname and the dictionary attr associated with it. If attr is omitted, an empty dictionary is created by default. Each new entry also gets a unique integer id which is generated.

  Example:
  `db.add_person(‘Ershaad Basheer’, {‘address’:’SERC701F’, ‘ext’:4217})`

- `add_attribute(id, attrname, value)`
  Adds an attribute to an already existing entry. Prints an error message if the entry does not exist.
  Example: `add_attribute(1, ‘department’, ‘Math’)`

- `del_byid(id)`
  Remove entry with id. Print error message if entry does not exist

- `retrieve_names(name)`
  Return a list of (id,name) tuples that contain name (even partially)
  Example: `retrieve_names('Bashe')`

- `retrieve_attr(id)`
  Return a dictionary of attributes and their values for the entry with id

- Implement the appropriate special method `__len__()` so that passing a ContactDB type to the `len()` builtin function returns the number of entries

- Implement the appropriate special method `__contains__(x)` so that the '`in`' keyword checks if a (partial) name is present among the names in the DB
  Example:
  ```
  >>> print('Ersha' in mycontacts)
  True
  ```

Q3

Create a program which uses the class from Q2 and manages a list of people. It should provide an interface similar to the following:

```
Welcome to ContactsDB!

Available commands:
list                  list persons in contact database
add                   add person into contact database
remove [person ID]    remove person with a given ID
details [person ID]   show full details of a given person
search [name]         search for a person with a given (partial) name
                      and list matches

add_attr  [person ID]            add attribute to person

Command> add

Enter Full Name: Ershaad Basheer

Command> list

1: Ershaad Basheer

Command> add_attr 1

Enter attribute name: Department
Enter value: Math

Command> details 1

Ershaad Basheer
Department: Math

Command> search Basheer
1: Ershaad Basheer
```

Bonus Points: Add a save command which saves the list of persons to a file. If you close the program and start it again, it should try to read that file and recreate the list of people. If the file doesn’t exists, start with an empty person list.