Programming I

AUTUMN ASSIGNMENT #1

This program assignment shall be an improvement on a previous lab exercise in Topic 3. It is designed to assess skills gained in functions, iteration and sequence types as well as parsing arguments given from the terminal at time of execution.

Activity

Using the Python you have learned to this point, and on the understanding above, write a program called "**pinger2.py**".

Create a program to behave as shown below using a standard template. Use the "European Union Public Licence v1.2" or any other Public licence you prefer.

- Create functions to manage **`ping`** and **`licence`** information.
- Create a **`main()`** function that is called once it is verified that the program is running its own instance and is not being called by a different program.
- Any IP addresses received by the program, either as terminal arguments or from a prompt MUST be verified as legitimate IP addresses.
- Program functionality should confirm that supplied IP addresses are **`alive`** or **`unreachable`**.

This program should request an IP address as input from the user, test connectivity to it and as illustrated below, return a message to the terminal confirming the status of the address as either alive or unreachable.

The **-h** or **--help** switch to return a helpful message.

```
~$ ./pinger2.py -h
usage: pinger2.py [-h] [-l] [-v] [-i [IP [IP ...]]]

pinger2.py IP address pinger program

optional arguments:
  -h, --help            show this help message and exit
  -l, --licence         pinger2.py licence information
  -v, --version         pinger2.py version information
  -i [IP [IP ...]], --ip [IP [IP ...]]
                        IP address list

~$ ./pinger2.py --help
usage: pinger2.py [-h] [-l] [-v] [-i [IP [IP ...]]]

pinger2.py IP address pinger program

optional arguments:
  -h, --help            show this help message and exit
  -l, --licence         pinger2.py licence information
  -v, --version         pinger2.py version information
  -i [IP [IP ...]], --ip [IP [IP ...]]
                        IP address list
```

Return the value of the **__author__**, **__copyright__** and **__licence__** dunders to the user upon a **-l** or **--licence** switch. Accept misspelling of the word licence as license too.

```
~$ ./pinger2.py -l

Author    : Diarmuid O'Briain
Copyright : Copyright 2021, Institute of Technology Carlow
Licence   : European Union Public Licence v1.2

~$ ./pinger2.py --licence

Author    : Diarmuid O'Briain
Copyright : Copyright 2021, Institute of Technology Carlow
Licence   : European Union Public Licence v1.2

~$ ./pinger2.py --license

Author    : Diarmuid O'Briain
Copyright : Copyright 2021, Institute of Technology Carlow
Licence   : European Union Public Licence v1.2
```

Return the value of the **__version__** dunder to the user upon a **-v** or **--version** switch.

```
~$ ./pinger2.py -v

Version: v1.0

~$ ./pinger2.py --version

Version: v1.0
```

Having entered nothing present a prompt to the user and for the IP address supplied, return **alive**|**unreachable**|**Error** message.

```
~$ ./pinger2.py

Enter an IP address: 192.168.0.1
192.168.0.1 is alive
```

Enter a list of IP address for testing with the **-i** or **--ip** switch. The program tests each and returns **alive**|**unreachable**|**Error** message.

```
~$ ./pinger2.py -i 192.168.0.1 265.3.2.2 8.8.8.8 1.1.1.1 192.168.0.23

192.168.0.1 is alive
265.3.2.2 - ERROR: is not a properly formatted address
8.8.8.8 is alive
1.1.1.1 is alive
192.168.0.23 is unreachable

~$ ./pinger2.py --ip 192.168.0.1 265.3.2.2 8.8.8.8 1.1.1.1 192.168.0.23

192.168.0.1 is alive
265.3.2.2 - ERROR: is not a properly formatted address
8.8.8.8 is alive
1.1.1.1 is alive
192.168.0.23 is unreachable
```

Submission

Submit a report (no more that 1,500 words) that describes how you carried out this assignment, any difficulties you encountered. The report to be submitted in PDF format (It will NOT be accepted in any other format). The report should have the following sections:

1. INTRODUCTION
2. ACTIVITY
3. ANALYSIS & INTERPRETATION OF ACTIVITY
4. CONCLUSIONS
5. REFERENCES
6. APPENDICES
    - Test showing examples of the code running

In terms of formatting etc.. use a project report template from here:
https://www.obriain.com/training/thesis/

Put the report, the Python code file and a text file showing the terminal output as you run the program in a directory called "C00<number>_AUTUMN_ASSIGNMENT", where C00<number> is replaced with your student number, i.e. C00123456.

Compress this folder into any of the following formats **`.tgz, .tar.gz, .bz2, .zip`**. It should have a name such as "C00<number>_AUTUMN_ASSIGNMENT.tgz, i.e. **`C00123456_AUTUMN_ASSIGNMENT.bz2`**

Upload the resultant compressed file.

NOTES:

- SUBMISSIONS WILL NOT BE ACCEPTED AFTER THE DUE DATE
- SUBMISSIONS WILL NOT BE GRADED IF FILES ARE UPLOADED IN THE WRONG FORMAT