Closed Lab Solution

- Review Closed Lab 01
Preliminaries: Additional List Operations

- List membership: \texttt{in}
  - returns Boolean
  - \( e \text{ in } L \)
- List of consecutive integers: \texttt{range}
  - returns a list
  - \texttt{range(10)} \[ 0, 1, \ldots, 9 \]
  - \texttt{range(100, 200)} \[ 100, 101, \ldots, 199 \]
- Assignment: \texttt{=}
  - creates an alias

Preliminaries: Random

- \texttt{random} provides a number of helpful methods
  - \texttt{random()} returns a float in \([0, 1)\)
  - \texttt{randint(x, y)} returns an int in \([x, y]\)
  - \texttt{shuffle(L)} permutes \(L\) in place
- Multiple ways to import
  - \texttt{import random}
  - \texttt{import random as rd}
  - \texttt{from random import random, randint, shuffle}
Iteration

- **while loops**
  - similar to other languages with minor syntactic differences

- **for loops**
  - primarily list based

```
i = 0
while i < 10:
    instructions
    i += 1
```

For loops with range and `len`

- `range(len(L))` returns list with an int value, beginning with 0, for each element in a list `L`.
- **iterator**
  - operate on each element of list
  - `e` takes value of each element in list in turn

```
for i in range(len(L)):
    instructions
```

```
for e in L:
    instructions
```
Important note about iterators

- Consider code this code fragment
  ```python
  for e in L:
    e = 2 * e
  ```

  - `e` is really a reference to each element in list
  - `e = 2 * e` reassigns the reference but doesn’t affect the value stored in the list
  - Thus `L` is unchanged

A big BUT concerning previous note

- Let `L` be a list of lists
  ```python
  for e in L:
    e[0] = 2 * e[0]
  ```

  - `e` is a reference to each element in list
  - `e[0] = 2 * e[0]` does not reassign the reference
  - Thus `L` is updated