Outline for October 23, 2024

Reading: §4.5, 7 **Due:** Homework 2, due October 30, 2023

1. Using floating point numbers [*roundoff.py*]
2. Using random numbers
   (a) `import random`
   (b) Problem: compute π by tossing darts at a unit square
   (c) First build routine to simulate dart toss at unit square [*mc-1.py*]
   (d) Then build routine to see if co-ordinates are in unit circle [*mc-2.py*]
   (e) Then build routine to read in number of tosses [*mc-3.py*]
   (f) Put it all together [*mc-4.py*]
   (g) Graphics! [*mc-5.py*]
3. Files
   (a) What is a file?
   (b) What can you do with it? (For example, read, write, append)
   (c) Types of files (text, binary)
4. File Input and Output for text files
   (a) Opening and closing: `open(filename, mode)`, `close()`
   (b) Reading: `readline()`, `readlines()`, `read()`, `read(n)`
   (c) Writing: `write(str)`, `writelines(list)`
5. Exception `EOFError` — input function encounters an end of file
6. Examples
   (a) Print out a named file [*fileio1.py*]
   (b) Print out a named file and prepend line numbers [*fileio2.py*]
   (c) Store the output in *filename*.lst [*fileio3.py*]
7. Examples
   (a) Put lines in a file in random order [*randlines.py*]
   (b) Read in a list of words from a file, then search it as requested; similar to linear search program [*search-1.py*]
   (c) Now see how many words you checked total [*search-1c.py*]
8. Dictionary
   (a) Collection of key-value pairs
9. Creating dictionaries
   (a) Using `d = {}`
   (b) Using `d = dict()`
10. Methods for dictionaries
    (a) `k in D`: True if dictionary `D` has key `k`; else False
    (b) `D.keys()`: list of keys in `D`
    (c) `D.values()`: list of values in `D`
    (d) `D.items()`: list of tuples (key, value) in `D`
    (e) `D.get(k, d)`: if key `k` in `D`, return associated value; else return `d`

(f) `del D[k]`: delete tuple with key `k` from `D`

(g) `D.clear()`: delete all entries in `D`

11. Example: memos

(a) Remember how slowly the recursive Fibonacci number program *rfib.py* ran? Here is a faster recursive version that uses memos [*rfibmemo.py*]