Programming Exercise 5

File Viewer

In this exercise you will use **command-line arguments** and **file positioning** to provide a file viewer to the user. We are going to assume that you have a very small amount of memory and you will only be able to store one screen of text from the file at a time. You will be rereading portions of the file to display the appropriate text on demand. Implement this functionality in a function declared as

```
    def view(fname, view_size=25): ….
```

It will be necessary for you to record the positions in a text file where a screenful begins. First, traverse the requested file to determine the file positions where each portion of **view_size** lines begins (in other words, read **view_size** lines, then record the file position, then repeat this process until you've passed through the entire file once; record file positions by calling **file.tell**). Then display the first page, and give the user the following prompt:

```
Command [u,d,t,b,#,q]:
```

The commands are interpreted as follows:

u	move **u**p one page; if already at the top, wrap to the last page
d	move **d**own one page; if at the bottom, wrap to the first page
t	move to the **t**op (first) page
b	move to the **b**ottom (last) page
#	moves to page number (1-based)
q	**q**uit

If the user just hits the Enter key, consider that a "**d**own" command. If the user enters a number, move to that page/screen. If the number is out of range, move to the top or bottom, as appropriate. You will use **file.seek** to move from page to page. Pad the display with extra newlines on the last page as needed so it fills **view_size** lines of screen space.

Obtain the file name from the command line when *view.py* is launched as the top-level module. In addition, process an optional, second command-line argument that represents the desired screen display size (which defaults to 25, as shown above). For example,

```
$ python3 view.py yankee.txt 20
```

will allow viewing the text in the file **yankee.txt** 20 lines at a time. Command-line arguments are obtained with **sys.argv**. Do not read the entire file into memory. Read it a line at a time and only hold a screen at a time in memory. Don't allow viewing empty files.

Use the file **yankee.txt**, found in this assignment. A sample user session is provided in the file **sample.txt**.

The **tell** file method returns a value that represents a file position. To return to that position later, you call **seek**:

```
pos = f.tell()
…
f.seek(pos)
```

You can read more about these methods in the documentation on python.org.