CS106A Final Exam Reference

Common Python List Functions:

`alist.append(x)`	Appends `x` to the end of `alist`
`alist.insert(index, x)`	Inserts `x` into `alist` at `index` After the insert function, x will be at index
`alist.pop()`	Removes the last element of `alist`
`alist.sort()`	Sorts the element of `alist` into ascending / alphabetical order
`len(alist)`	Returns the number of elements (length) of `alist`
`alist[index1:index2]`	Returns a new list from `index1` up to but not including `index2`

Graphics commands:

`canvas = Canvas(width, height, 'Name')`	Creates a new drawing canvas with a width of `width` pixels and a height of `height` pixels with a title of `Name`
`canvas.create_line(x1, y1, x2, y2, color)`	Draws a line from (`x1`, `y1`) to (`x2`, `y2`) in `color`
`canvas.create_rectangle(x1, y1, x2, y2, color)`	Draws a rectangle bounded by (`x1`, `y1`) and (`x2`, `y2`) in `color`
`canvas.create_oval(x1, y1, x2, y2, color)`	Draws an oval bounded by the rectangle (`x1`, `y1`) and (`x2`, `y2`) in `color`.
`canvas.create_text(x, y, anchor=anchor, font=font, font_size=font_size, text='some_text', color=color)`	Draws the text `text` at position (`x`, `y`) in `font` with `font_size` in `color`, anchored by `anchor`. `anchor` can be one of **`"n"`, `"ne"`, `"e"`, `"se"`, `"s"`, `"sw"`, `"w"`, `"nw"`, or `"center"`**, that describes what side of your text you want to position at the (`x, y`)

`canvas.move(obj, dx, dy)`	Moves **obj** by adding **dx** to its current **x** value, and by adding dy to its current **y** value.
`canvas.moveto(obj, x1, y1, x2, y2)`	Moves **obj** to location (`x1`, `y1`) and (`x2`, `y2`)

Sorting

```
sorted(alist, key=None, reverse=False)
```

Returns a new list that is sorted according to the `key` function and reversed if `reverse` is `True`.

File reading

Given an open file, `f`, the following can functions can be used on `f`:

`f.read()`
Reads in the entire file, returning a string

`f.readline()`
Reads the next line in the file, returning a string

`f.readlines()`
Reads all of the lines in the file, returning a list of strings

`for line in f:`
Reads one line at a time from the file, where `line` will be the line read. When all lines have been read, the loop ends.