CS150A & CS150B Introduction to Computer Science (Zaring)
Fall 2018
Lab 7

(1) Create a folder named `Lab07` on your desktop and then download all the files from the `Lab 7` item on the `Labs` page of the CS150 Katie course to your `Lab07` folder. The rest of this lab won't work unless you download all the files and download them into the folder `Lab07`.

(2) Using Wing 101, consider the program in the file `Lab07Part03.py`:

- The program displays the bear logo. Run the program to verify this.
- Modify the program so that it also shows the pictures in the files `Eniac.gif` and `Olin.gif`, each in a separate window, so that you can see all three images on the screen at the same time. Make it so that clicking on an image causes it to close.
- Don't just close your image-windows by clicking on the little X in the upper-left corner of the image-window: this can cause your program problems later on.
- If you execute the `exitonclick` methods for, say, image-windows A, B, and C (in that order: A then B then C), you have to click on the windows in that same order (i.e., A then B then C) when closing them.
- Insert a comment containing your first and last names at the top of your program and submit your completed file `Lab07Part03.py` using the `Lab 7` item on the `Labs` page of the CS150 Katie course.

(3) Consider the program in the file `Lab07Part04.py`:

- The program displays the bear logo. Run the program to verify this.
- Modify the program so that it instead prompts for and reads in (from the user) the name of an image file and then displays the image from the file having that name.
- Insert a comment containing your first and last names at the top of your program and submit your completed file `Lab07Part04.py` using the `Lab 7` item on the `Labs` page of the CS150 Katie course.

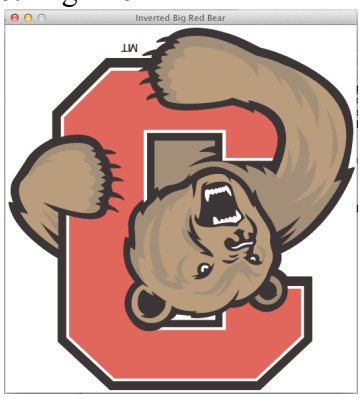
(4) Consider the program in the file `Lab07Part05.py`:

- The program displays the bear logo and then produces and displays a dimmer version of the bear logo. Run the program to verify this.
- Modify the program so that instead of producing and displaying a dimmer version of the bear logo, it produces and displays a version of the bear logo with all the greenness and all the blueness removed from all of the pixels. That is, it should ultimately show images something like

  both showing on the screen at the same time.
- Insert a comment containing your first and last names at the top of your program and submit your completed file `Lab07Part05.py` using the `Lab 7` item on the `Labs` page of the CS150 Katie course.

(5) Consider the programs in the file `Lab07Part06.py`:

- The program displays the bear logo and then produces and displays an exact copy of the bear logo. Run the program to verify this.
- Modify the program (by replacing the definitions of the function `duplicateImage` with the definition of a new function `invertedImage`) so that instead of producing and displaying an exact copy of the bear logo, it produces and displays an inverted (i.e., upside-down) version of the bear logo. (See the next item in these instructions for information on inverting images.) Your program should ultimately show images something like



both showing on the screen at the same time.

- Inverting an image is conceptually simple: the pixels on row 0 (i.e., the top row) in the original image go on the bottom row of the inverted image, the pixels on row 1 in the original image go on the next-to-bottom row of the inverted image, the pixels on row 2 in the original image go on the next-to-next-to-bottom row of the inverted image, … , and the pixels on the bottom row in the original image go on row 0 (i.e., the top row) of the inverted image. You main job is to answer the question "the pixels on row *j* in the original image should go on *what* row of the inverted image?"
- Insert a comment containing your first and last names at the top of your program and ubmit your completed file `Lab07Part06.py` using the `Lab 7` item on the `Labs` page of the CS150 Katie course.