ETEC3702 – Concurrency
Lab 5 – Improving Performance of I/O-Bound Processes

Due date: 16 February by the end of the day.

For this lab we are going to write a program that downloads 10 images from NASA. The 10 images are located at these URLs:

```
fileList="images-assets.nasa.gov/image/PIA04921/PIA04921~orig.jpg",
            "images-assets.nasa.gov/image/PIA10600/PIA10600~orig.jpg",
            "images-assets.nasa.gov/image/GSFC_20171208_Archive_e001925/
GSFC_20171208_Archive_e001925~orig.jpg",
            "images-assets.nasa.gov/image/PIA11999/PIA11999~orig.jpg",
            "images-assets.nasa.gov/image/GSFC_20171208_Archive_e001262/
GSFC_20171208_Archive_e001262~orig.jpg",
            "images-assets.nasa.gov/image/PIA15536/PIA15536~orig.jpg",
            "images-assets.nasa.gov/image/PIA00583/PIA00583~orig.jpg",
            "images-assets.nasa.gov/image/GSFC_20171208_Archive_e001762/
GSFC_20171208_Archive_e001762~orig.jpg",
            "images-assets.nasa.gov/image/PIA05185/PIA05185~orig.jpg",
            "images-assets.nasa.gov/image/PIA03225/PIA03225~orig.jpg"]
```

Function 1:

Write a function called `downloadSequential(filesToDownloadList)` that sequentially downloads each of the files in the list passed to the function. The downloads should be performed sequentially one after the other. The downloaded files should each be stored in local files with the same names as the original files. Your program should display how long each file took to download and should display the overall time that the entire program took to execute. Return the overall download time.

Hint: I'd suggest using the python `urllib.request` module. It works as follows:

```
import urllib.request

urllib.request.urlretrieve("http://"+url_of_file, output_file_name)
```

Note that the output filename should not include the entirety of the URL. You may want to consider using split( ) to get just the filename portion of the URL from the end of the URL string.

Function 2:

Write a second function called `downloadConcurrent(filesToDownloadList)` that uses a separate thread for each of the file downloads. Be sure that this function also times how long each file download takes and the overall time needed to start all of the threads and wait for them to complete all downloads. Return the overall download time.

Main Program:

Write a main program that calls the sequential version of the function followed by the concurrent version of the function. Use the returned processing times to compute and display the speedup achieved.

What was the calculated speedup?
How much faster / slower was the concurrent version?