I. Concurrency

1. Write a function **`getpositions(fnom,chunksize)`** where **`fnom`** is the name of a file and **`chunksize`** is a positive integer. The function should return a list of file positions that would divide the file contents into segments of length **`chunksize`**, with the possible exception that the last segment could have a positive size less than **`chunksize`**.

2. Write a function **`chunk_counter(f,pos,csize,b)`**, where **`f`** is a function object and **`csize`** is a positive integer, that returns the number of occurrences of byte **`b`** in the segment of **`f`** starting at position **`pos`** and having length at most **`csize`**.

3. Write a function **`total_counter(fnom,b)`** that returns the total number of occurrences of byte **`b`** in the file named fnom. Your function should use multithreading with and use the functions **`getpositions`** and **`chunk_counter`**; assume that your processor has 8 cores.

4. Would using multiprocessing instead of multithreading run slower or faster than the multithreaded version?

II Regular Expressions

5. For each of the following create a single regular expression that:

(a) recognizes the following strings: “bat”, “bit”, “but”, “hat”, “hit”, or “hut”.

(b) matches any word and single letter separated by a comma and single space, as in last name, first initial.

(c) matches a one or two digit number string representation of a month of the year (January,…, December).