/\*Assumption:

\* 1) N-ary Tree where Folders act as the vertices or Nodes and they contain or point to a list of Files

\* 2) Files are in an iterable array of some sort with a known value for size or length.

\*

\* int Total(root) //start at the root directory

\* int total = 0;

\* if(ArrayList<File> exists){ total += get the total number of Files}

\* if(Folder exits, total += Total(Folder)

\* return total;

\*

\* If item is a file, Loop and count it and add it to the total to be returned.

\* Note(iterative loops keep overhead down [no Stack memory allocation])

\* If item is a Folder, recurse it and return total items found

\* [(this will repeat the above steps) Upon the return, a

\* value will be passed back from that function call on the

\* Stack to the preceding function on the Stack and be added to it.]

\*

\* This process will repeat until each folder and file is counted, returning a grand total

\*/