```
import os

# represents folder
class Folder:

    # initialize folder name and parent folder; None for root folder
    def __init__(self, name, parentFolder):
        self.name = name
        self.parent = parentFolder
        if parentFolder!=None:
            self.path = self.parent.path+"/"+name
        else:
            self.path = self.name
            self.parent = "0"

    # files and folders are lists of File and Folder objects; computs size of this Folder
    def setChildren(self, files, folders):
        self.files = files
        self.folders = folders
        self.size = 0
        for f in files:
            self.size += f.size
        for f in folders:
            self.size += f.size

    # output data entry to tsv file
    def populateTSV(self, resultFile, fid=1):
        self.fid = fid
        resultFile.write(self.name+"\t"+str(self.size)+"\tY\t"+str(int(str(self.parent))-1)+"\t"
        +str(long(os.path.getmtime(self.path)))+"\n")
        for f in self.files:
            fid+=1
            f.populateTSV(resultFile)
        for f in self.folders:
            fid+=1
            fid = f.populateTSV(resultFile,fid)
        return fid

    def __repr__(self):
        return str(self.fid)

# represents File
class File:

    # initialize name and parent folder
    def __init__(self, name, parentFolder):
        self.name = name
        self.parent = parentFolder
        path = self.getPath()
        self.size = os.path.getsize(path)

    # path is path of folder + name of file
    def getPath(self):
```

```
        return self.parent.path+"/"+self.name

    # output data entry to tsv file
    def populateTSV(self, resultFile):
        resultFile.write(self.name+"\t"+str(self.size)+"\tN\t"+str(int(str(self.parent))-1)+"\t"
        +str(long(os.path.getmtime(self.getPath())))+"\n")

# from a root folder name, create a tsv file containing all files and folders
def getFileStructure(rootFolderName, resultFileName="results.tsv"):
    resultFile = open(resultFileName, "w")
    resultFile.write("name\tsize\tisfolder\tparent\tmoddate\n")
    root = getFileStructureHelper(rootFolderName, None)
    root.populateTSV(resultFile)
    resultFile.close()


def getFileStructureHelper(folderName, parentFolder):
    currFolder = Folder(folderName, parentFolder)

    # gets list of names of all files and folders one level deep in folder, prepends folder name
    filesFolders = os.listdir(currFolder.path)

    # splits list into files and folders; file names gets converted to (file name, file size)
    tuples
    files = map(lambda f:File(f, currFolder), filter(lambda f:not os.path.isdir(currFolder.path+
    "/"+f), filesFolders))
    folders = map(lambda f:getFileStructureHelper(f, currFolder), filter(lambda f:os.path.isdir(
    currFolder.path+"/"+f), filesFolders))

    currFolder.setChildren(files, folders)

    return currFolder


# folderName is "test" which is included with FileFinder.py
# You can try on different folders, but be careful of messing with whole system drives
getFileStructure("test")
```