**Portability**

It should be possible to run your program on any operating system — Windows, macOS, Linux, etc. — so long as there is a Python 3.8 implementation for it. By and large, this doesn't require much special handling: Python is already reasonably independent of its platform. However, since you're dealing with a filesystem, you do have to be cognizant of how filesystems are different from one operating system to another. Most notably, paths are written differently:

* On Windows, paths tend to be a *drive letter*, followed by a colon, followed by a sequence of directory names separated by backslashes, such as **D:\Docs\UCI\ICS32\Homework**.
* On macOS, Linux, and other flavors of Unix, paths tend to be a sequence of directory names preceded by forward slashes instead, such as **/home/student/uci/ics32/homework**.

The way to isolate this distinction is to find tools in Python's standard library that isolate them for you. Don't store paths as strings; store them as path objects instead. (More on this below.)

**Handling failure**

You'll want to handle failure carefully in this program. In general, your program should not crash just because one activity fails; it should instead quietly skip the offending file or directory and continue, if possible. For example:

* If, during the search, accessing some directory fails, the search should still continue attempting to access other directories.