

---

## User filenames

---

Your company had some data entry work that needed doing, and they contracted the job out. The work consisted of labelling input data files. The data entry workers were given a filename specification to follow. The data entry workers however were not computer scientists, and, being human, made mistakes, misunderstood the specification, and made erroneous assumptions. You've now been brought in to clean things up. Fun fact: the author of this étude had to do this exact task themselves.

---

### Task

Write a program that accepts a directory name as a command line argument. Within that local folder will be a series of plaintext files, and potentially other subdirectories which themselves may contain files and subdirectories. Your program should walk this directory structure, assemble a list of files to be read in sequential order, read the contents of those files, and save the concatenated output to a single file. The subdirectory names have no bearing on the file ordering or numbering.

#### 1.1 Filename specification

The following filename specification was provided to the data entry workers.

- All filenames must end with '.txt'
- Each filename consists of three pairs of two-digit identifiers, each separated by a single hyphen
- The first digit pair refers to job site (range 01 to 05), the second pair to lab desk (range 01 to 25), the third pair to job number (range 01 to 99).
- All job numbers must be sequential, with no missing job numbers. That is, 01 then 02, ..., up to 99.

Filename example: 03-24-43.txt Refers to: Job site 3, lab desk 24, job number 43.

#### 1.2 Organizing the files

Your program should organize the list of files into a sequential order, ignoring which subdirectory they were found within. Files should be ordered first by job site, then lab desk, then job number. The following example is the only valid order for the given filenames: 01-01-01.txt, 01-01-02.txt, 01-02-01.txt, 01-02-02.txt, 04-01-01.txt, 04-02-01.txt.

### 1.3 Program output

Once the ordered list of filenames is created, your program should read each of the files in order, and concatenate file contents into one plaintext output file called 'result.txt'. The contents of each new file should begin on a new line. For example, if the first file contained 'Fail' on two separate lines, while the second file contained only a single 'Pass', your output would look like:

```
Fail
Fail
Pass
```

### 1.4 Handling invalid cases

All files must be read, even if a filename violates the specification. Your job is to deduce what the correct order of files should be, given the filename specification. All files are readable and contain simple plain text (usually only one or two short lines).

---

### Relates to Objectives

1.2 1.3 1.4 2.1 2.2 2.5 2.6 2.7 2.9 3.1 3.5

(1 point, Pair)