

[Prompt #1: Range Interpreter](#)[Prompt #2: File Search](#)[Guidelines](#)[Submission](#)[sample-data.ps1](#)[sample-data.sh](#)[Solution: Range Interpreter](#)[Solution: File Search](#)

## File Search

Build a program which will find filenames in a folder/directory structure based on user input. Program must present a menu of files matching the search criteria and display the contents of a user selected file.

Judges will provide a folder structure and generic text files for testing.

- `<root path>/competition/milestone1/...` is for testing your code in the first milestone.
- `<root path>/competition/milestonex/...` is for testing your code in all other sections.

## Folder Structure Build Scripts

- [Windows PowerShell Script](#)
- [\\*NIX BASH Script](#)

## Milestone 1 - Basic Search

Build one or more functions which solve the core search problem. *Minimum* required input:

1. Starting folder/directory path (str)
2. Search term (str) which will be in a filename

Basic search must check the starting folder/directory and all subfolders/subdirectories. It must return an array (or your language's equivalent) of pathnames of all successful matches.

**Restriction:** OS Native mechanisms or libraries which allow file system access are permitted, but *search* features are off-limits.

Program must show steps required to examine folder/directory contents, enumerate child directories, and continue the search into the child and grandchild directories.

## Milestone 2 - Constrained Search Depth and Infinite Loop Prevention

Add features to the program:

1. Allow the user to specify the search depth in the folder/directory structure.
2. Detect Microsoft Windows junctions or \*NIX symbolic links to prevent infinite loops

Search depth parameters and defaults:

- Negative one ( `-1` ) indicates no restriction and must be the default value.
- Zero ( `0` ) indicates search should only occur in the user specified folder/directory.
- Positive integer of one ( `1` ) or greater indicates desired search depth.
- Other values must provide meaningful error message

## Milestone 3 - End User Ready Tool

Create a CLI or GUI tool. Tool must provide the following:

- Syntax or usage help.
- Validate input and provide meaningful error messages.
- Upon completion of search, present user with a menu of matching filenames (and paths) and offer the ability to view the file contents.

## Milestone 4 - Power Features

Add one or more of the following features.

- Run continuously and provide these 2 features.
  - View another file from the current search.
  - Alter search parameters and view different files.
- Save parameters of current session for future sessions.
- Save search results.
- Demonstrate Windows, Linux, MacOS cross-platform functionality.
- Add unit testing for CI/CD workflows.
- Establish a local Git repository and demonstrate diffs for each milestone.

Show Details

Last Modified 2024-04-04 09:58 PST