



Assignment 2: Cycle Finder

This assignment asks you to write a cycle-finding application. Systems that support symbolic links open up the possibility of cycles in the directory hierarchy. It is useful to be able to discover the cycles in the hierarchy. Many commands detect these cycles but do not report them, so having a command that can list them would be beneficial.

Your program must implement the command described by the man page below. You can write it in C or C++. You cannot use any commands to solve this problem; you may use any tree traversal method you choose.

NAME

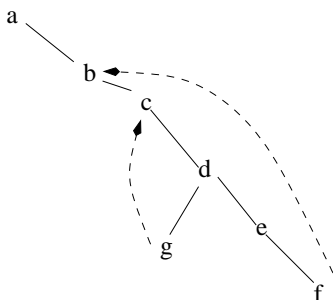
```
cyclefind - find all cycles in a file hierarchy
```

SYNOPSIS

```
cyclefind [directory_name] ...
```

DESCRIPTION

`cyclefind` recursively searches through the directory hierarchies whose roots are the directories specified on the command line, writing all simple cycles that it finds to standard output. If no directory is specified, the current working directory is assumed. A cycle is simple if all file names in it are unique except the first and last. For example, in the graph below,



there is a link from `f` back to `b` and a link from `g` back to `c`. The cycle `b-c-d-e-f-b` is simple and `c-d-g-c` is simple. The command

```
cyclefind a
```

will report this set of cycles, as will

```
cyclefind d
```

showing that `cyclefind` reports cycles that contain the root of the search. `cyclefind` displays the cycle as a sequence of file names, one per line, terminated by a blank line. The files that are symbolic links to other files are noted in the output with the word “symlink”. There is no rule as to which filename appears first in the cycle.

`cyclefind` crosses mount points to search for cycles across file systems. If it tries to follow a broken link, it silently ignores it.



EXIT STATUS

- 0 If it succeeded.
- 1 If it failed.

Submitting the Assignment

Create a directory in

```
cs82010.gc.cuny.edu:/home/class_stuff/cs82010/projects/project2
```

named `username_hwk2`. Give it permissions 700 so that only you have access to it. Within that directory, put all project-related files. For example, if you have a multi-file project with a Makefile, all files should be there. If you use a Makefile, make sure you name the executable `cyclefind` when it is built. If you write a multi-file program, you must create a Makefile, so that I can compile and build it with the single command "make." If it is a single file, name it `cyclefind.X`, where `X` is the GNU extension for the language (`.c` for C, `.cpp` or `.C` for C++, etc.)

The program must be well-documented and must contain a preamble or prologue at the very top like the ones in my demo programs, which provides authorship, use, and other pertinent information. All functions should have documentation that details what their parameters are and what its return value is.