VFY.rb Test cases

For each situation test all perms of command options.
- also each in root / non-root.

Symlinks

1. Orig /foo/a.txt, b.txt
   back/foo → sym to orig/foo
   ⇒ Fail
   - Counts a.txt, b.txt

2. — Same as above but swap orig_back
   ⇒ Fail
   - Does not count a.txt, b.txt

3. orig/bar.txt
   back/bar.txt → sym to orig/bar.txt
   ⇒ Fail

4. Same as 3 but swap orig↔back
   ⇒ Fail.

5. orig/foo → sym to /foo
   back/foo → sym to /bar
   ⇒ Fail

when both foo dir and file.

6. (when /foo both dir and file)

orig/foo -> /foo
back/foo -> /foo

=> Pass.

TODO: All of these with and without skip symlinks

Files + Dirs

(hidden)

7.
orig/foo (dir)
back/foo (file)
=> Fail

8.
orig/foo (file)
back/foo (dir)
=> Fail

9.
orig/foo (file)
back/foo (same file)
=> Pass

10. orig/foo (file with "ABC")
back/foo (file with "DEF")
=Fail.

11. orig/.foo
back/.foo
(foo both file and dir, same)
=> Pass

12. Same as 11 but diff contents
=> Fail.

13.
orig/foo
back/ (no foo)
=> Fail
(for both dir and file foo)

Root directory

14. – orig path is symlink
    – backup is not
    ⇒ PASS

15. Inverse of 14. (PASS).

16. orig and back are the same path
    ⇒ PASS

17. orig and back are different paths, but these paths are symlinks to the same path.
    ⇒ PASS

Stay on filesystem

18. orig/
    back/foo → diff FS
    ⇒ Fail (dir should exist in orig)

19. Inverse of 18. (FAIL)

20. orig/foo/a.txt (on diff FS at foo/)
    back/foo (empty dir)
    ⇒ PASS if stay on FS
    FAIL if traverse FS

21.
orig/Foo/a.txt (diff fs at Foo/)
back/foo/b.txt
⇒ FAIL (foo should be empty).
(or should it?)

22.
orig/foo → diff fs
~~back~~/Foo → save FS, save contents
PASS if trees
FAIL if not (see #21).

23.
orig/foo → diff fs
back/foo → diff fs (not same as orig/foo)
⇒ PASS.

More?

Extra

- Combinators of differences to make sure the counts are correct.
  → Without more than one, it's just always adding to 0