Write a program which receives on the command line two arguments:

- The first is the name of a file, which contains a list of filenames and sha256 hashes, one per line, as in the following example:

```
05a229298a813b4976a4c976ee24ace54cfcb8667a4c6ed9d58cc3bdac29d5a1  FileA.pdf
301457bc8654dec78f9aec5768c59f795f24036cd35eb0801031408b21ca15e2  FileB.pdf
cb81a120d366a7a08d6941ade8ef9645b77704ee5769329152b28a59462f3df5  FileC.pdf
0cd23ee1853d2225189c734f0497a7d5d9de62e2c7379d868aeac51816867729  FileD.pdf
5d419afe116ac582cfe419222c21a988a467c9115467f82b6da2155d25760e6d  FileE.pdf
```

- The second parameter is a path (which must be a directory).

The program will then;

- Recursively scan this directory (the second arg.)
- For each file found, if the name matches one in the list (first arg.), it will compute its checksum, using the **`sha256sum`** program, reporting on stderr all inconsistencies (mismatches, files not found, etc)

Notes:

- **Use `fork()/exec*(), pipe()`** plus everything you feel appropriate.
- **You're not allowed to use temporary files.**
- **Use only syscalls for I/O**
  - open/close/read/write but **not** fopen/fread/fgets etc..

Optional:

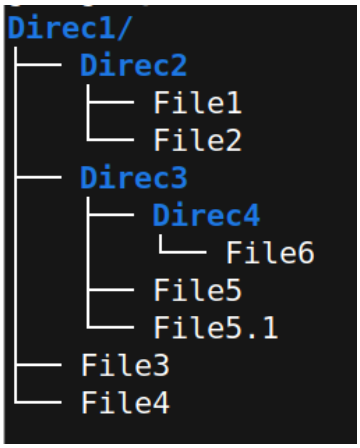
- For each subdirectory fork() a new child to scan the subtree, but limit the maximum number of parallel processes to **N** (a constant of your choice).
  Hint: probably you would need a couple of semaphores.

Example :

```
$ cat Sha256In

c991e9272396e7d1cb0bb092d656ebe5dfd141825967c9b284f30cba2634f0d4  File1
f35ac28a8507f030b7fa1937b9604648fdb54f1a7661256c89153cb6b4b038f8  File2
c244388c36db03afec72c9a56a5e4c30fe7c888126f78e52663a34024255a95e  File3
6bee0eed8899e224d2c85d71bab613b34313c71c1c42b65078d57791da45805f  File4
ec65a9da53aaa27333e8c55bf1f7afe34279c1ec1890120a9ba651b8b5869d85  File5
61812891bba8e185b60a42c33a77293a990361b17d0913bde3b57d520f44fdc6  File6
```

With the following structure:



Then (assuming File4 was corrupted):

```
$ ./a.out Sha256In Direc1

Checking ..
+++                                                    File5.1: not found
---                                                      File4: has been tampered

$
```