Contest: A Better Checker

For this project, there is a contest, which will compare checkers that can handle these more challenging condition checks:

1. Each `..` entry in directory refers to the proper parent inode, and parent inode points back to it. If not, print `ERROR: parent directory mismatch.`
2. Every directory traces back to the root directory. (i.e. no loops in the directory tree). If not, print `ERROR: inaccessible directory exists.`

This better checker will also have to do something new: actually repair the image, in one specific case. Specifically, your task will be to repair the "inode marked use but not found in a directory" error.

We will provide you with an xv6 image that has a number of in-use inodes that are not linked by any directory. Your job is to collect these nodes and put them into the `lost_found` directory (which is already in the provided image under the root directory). Real checkers do this in order to preserve files that may be useful but for some reason are not linked into a directory.

To do so, you will need to obtain write access to the file system image in order to modify it. This repair operation of your checker program should only be performed when `-r` flag is specified:

```
prompt> xcheck -r image_to_be_repaired
```

In this repair mode, your program should **not** exit when an error is encountered, but rather continue processing. For simplicity, you can also assume there is no other types of error in the provided image. It should exit only after it has created an entry under the `lost_found` directory for every lost inode.

The contest will be judged based on whether all extra tests are passed. If they are, the winner will be given to the most readable implementation.