

# Using an oracle to see where your code is producing the wrong output

 devblogs.microsoft.com/oldnewthing/20250530-00/?p=111232

May 30, 2025



You may find yourself trying to generate some output, and the output is wrong, where “wrong” means “is not accepted by whatever program consumes that output.” It is often the case that you have a reference implementation, which I somewhat whimsically call an “oracle”, that is known to produce acceptable output. In that case, you can use that reference implementation to check your implementation.<sup>1</sup>

Give your implementation and the reference implementation the same input, and see if they produce the same output. For binary formats, you can use a binary file viewer program. In a pinch, you can use the Windows `comp.exe` or `fc.exe /b` programs. The files will not be identical,<sup>2</sup> and the tool will tell you where they differ. Work backward from that file offset to your program to see why your program chose to write the wrong data at that point. (Maybe you forgot to open the output file in binary mode?)

<sup>1</sup> I used a variation of this technique some time ago when [reverse-engineered the calling convention for various historical CPU architectures](#).

<sup>2</sup> If the file contents are identical at the byte level, yet one is accepted and the other is rejected, then the problem is not in the file contents. Maybe there is file metadata that does not match, like the [Mark of the Web](#).