

4. Exercises - Q3-8

3. Sometimes the additive and XOR-based checksums produce the same checksum (e.g., if the data value is all zeroes). Can you pass in a 4-byte data value (using the -D flag, e.g., -D a,b,c,d) that does not contain only zeroes and leads the additive and XOR-based checksum having the same value? In general, when does this occur? Check that you are correct with the -c flag.
4. Now pass in a 4-byte value that you know will produce a different checksum values for additive and XOR. In general, when does this occur?
5. Use the simulator to compute checksums twice (once each for a different set of numbers). The two number strings should be different (e.g., -D a1,b1,c1,d1 the first time and -D a2,b2,c2,d2 the second) but should produce the same additive checksum. In general, when will the additive checksum be the same, even though the data values are different? Check your specific answer with the -c flag.
6. Now do the same for the XOR checksum.
7. Now let's look at a specific set of data values. The first is: -D 1,2,3,4. What will the different checksums (additive, XOR, Fletcher) be for this data? Now compare it to computing these checksums over -D 4,3,2,1. What do you notice about these three checksums? How does Fletcher compare to the other two? How is Fletcher generally "better" than something like the simple additive checksum?
8. No checksum is perfect. Given a particular input of your choosing, can you find other data values that lead to the same Fletcher checksum? When, in general, does this occur? Start with a simple data string (e.g., -D 0,1,2,3) and see if you can replace one of those numbers but end up with the same Fletcher checksum. As always, use -c to check your answers.