**Programming Sucks**

A very friend I have with a job that involves picking up something heavier than a laptop more than twice a week eventually finds a way to slip something like this into conversation: "Bro,1 you don't work hard. I just worked a 4700-hour week digging a tunnel under Mordor with a screwdriver."

They have a point. Mordor sucks, and it's certainly more physically taxing to dig a tunnel than poke at a keyboard unless you're an ant. But, for the sake of the argument, can we agree that stress and insanity are bad things? Awesome. Welcome to programming.

**All programming teams are constructed by and of crazy people**

Imagine joining an engineering team. You're excited and full of ideas, probably just out of school and a world of clean, beautiful designs, awe-inspiring in their aesthetic unity of purpose, economy, and strength. You start by meeting Mary, project leader for a bridge in a major metropolitan area. Mary introduces you to Fred, after you get through the fifteen security checks installed by Dave because Dave had his sweater stolen off his desk once and Never Again. Fred only works with wood, so you ask why he's involved because this bridge is supposed to allow rush-hour traffic full of cars full of mortal humans to cross a 200-foot drop over rapids. Don't worry, says Mary, Fred's going to handle the walkways. What walkways? Well Fred made a good case for walkways and they're going to add to the bridge's appeal. Of course, they'll have to be built without railings, because there's a strict no railings rule enforced by Phil, who's not an engineer. Nobody's sure what Phil does, but it's definitely full of synergy and has to do with upper management, whom none of the engineers want to deal with so they just let Phil do what he wants. Sara, meanwhile, has found several hemorrhaging-edge paving techniques, and worked them all into the bridge design, so you'll have to build around each one as the bridge progresses, since each one means different underlying support and safety concerns. Tom and Harry have been working together for years, but have an ongoing feud over whether to use metric or imperial measurements, and it's become a case of "whoever got to that part of the design first." This has been such a headache for the people actually screwing things together, they've given up and just forced, hammered, or welded their way through the day with whatever parts were handy. Also, the bridge was designed as a suspension bridge, but nobody actually knew how to build a suspension bridge, so they got halfway through it and then just added extra support columns to keep the thing standing, but they left the suspension cables because they're still sort of holding up parts of the bridge. Nobody knows which parts, but everybody's pretty sure they're important parts. After the introductions are made, you are invited to come up with some new ideas, but you don't have any because you're a propulsion engineer and don't know anything about bridges.

Would you drive across this bridge? No. If it somehow got built, everybody involved would be executed. Yet some version of this dynamic wrote every single program you have ever used, banking software, websites, and a ubiquitously used program that was supposed to protect information on the internet but didn't.

**All code is bad**

Every programmer occasionally, when nobody's home, turns off the lights, pours a glass of scotch, puts on some light German electronica, and opens up a file on their computer. It's a different file for every programmer. Sometimes they wrote it, sometimes they found it and knew they had to save it. They read over the lines, and weep at their beauty, then the tears turn bitter as they remember the rest of the files and the inevitable collapse of all that is good and true in the world.

This file is Good Code. It has sensible and consistent names for functions and variables. It's concise. It doesn't do anything obviously stupid. It has never had to live in the wild, or answer to a sales team. It does exactly one, mundane, specific thing, and it does it well. It was written by a single person, and never touched by another. It reads like poetry written by someone over thirty.

Every programmer starts out writing some perfect little snowflake like this. Then they're told on Friday they need to have six hundred snowflakes written by Tuesday, so they cheat a bit here and there and maybe copy a few snowflakes and try to stick them together or they have to ask a coworker to work on one who melts it and then all the programmers' snowflakes get dumped together in some inscrutable shape and somebody leans a Picasso on it because nobody wants to see the cat urine soaking into all your broken snowflakes melting in the light of day. Next week, everybody shovels more snow on it to keep the Picasso from falling over.

There's a theory that you can cure this by following standards, except there are more "standards" than there are things computers can actually do, and these standards are all variously improved and maligned by the personal preferences of the people coding them, so no collection of code has ever made it into the real world without doing a few dozen identical things a few dozen not even remotely similar ways. The first few weeks of any job are just figuring out how a program works even if you're familiar with every single language, framework, and standard that's involved, because standards are unicorns.

**There will always be darkness**

I spent a few years growing up with a closet in my bedroom. The closet had an odd design. It looked normal at first, then you walked in to do closet things, and discovered that the wall on your right gave way to an alcove, making for a handy little shelf. Then you looked up, and the wall at the back of the alcove gave way again, into a crawlspace of utter nothingness, where no light could fall and which you immediately identified as the daytime retreat for every ravenous monster you kept at bay with flashlights and stuffed animals each night.

This is what it is to learn programming. You get to know your useful tools, then you look around, and there are some handy new tools nearby and those tools show you the bottomless horror that was always right next to your bed.

For example, say you're an average web developer. You're familiar with a dozen programming languages, tons of helpful libraries, standards, protocols, what have you. You still have to learn more at the rate of about one a week, and remember to check the hundreds of things you know to see if they've been updated or broken and make sure they all still work together and that nobody fixed the bug in one of them that you exploited to do something you thought was really clever one weekend when you were drunk. You're all up to date, so that's cool, then everything breaks.

"Double you tee eff?" you say, and start hunting for the problem. You discover that one day, some idiot decided that since another idiot decided that 1/0 should equal infinity, they could just use that as a shorthand for "Infinity" when simplifying their code. Then a non-idiot rightly decided that this was idiotic, which is what the original idiot should have decided, but since he didn't, the non-idiot decided to be a dick and make this a failing error in his new compiler. Then he decided he wasn't going to tell anyone that this was an error, because he's a dick, and now all your snowflakes are urine and you can't even find the cat.

You are an expert in all these technologies, and that's a good thing, because that expertise let you spend only six hours figuring out what went wrong, as opposed to losing your job. You now have one extra little fact to tuck away in the millions of little facts you have to memorize because so many of the programs you depend on are written by dicks and idiots.

And that's just in your own chosen field, which represents such a tiny fraction of all the things there are to know in computer science you might as well never have learned anything at all. Not a single living person knows how everything in your five-year-old MacBook actually works. Why do we tell you to turn it off and on again? Because we don't have the slightest clue what's wrong with it, and it's really easy to induce coma in computers and have their built-in team of automatic doctors try to figure it out for us. The only reason coders' computers work better than non-coders' computers is coders know computers are schizophrenic little children with auto-immune diseases and we don't beat them when they're bad.

**A lot of work is done on the internet and the internet is its own special hellscape**

Remember that stuff about crazy people and bad code? The internet is that except it's literally a billion times worse. Websites that are glorified shopping carts with maybe three dynamic pages are maintained by teams of people around the clock, because the truth is everything is breaking all the time, everywhere, for everyone. Right now someone who works for Facebook is getting tens of thousands of error messages and frantically trying to find the problem before the whole charade collapses. There's a team at a Google office that hasn't slept in three days. Somewhere there's a database programmer surrounded by empty Mountain Dew bottles whose husband thinks she's dead. And if these people stop, the world burns. Most people don't even know what sysadmins do, but trust me, if they all took a lunch break at the same time they wouldn't make it to the deli before you ran out of bullets protecting your canned goods from roving bands of mutants.

You can't restart the internet. Trillions of dollars depend on a rickety cobweb of unofficial agreements and "good enough for now" code with comments like "TODO: FIX THIS IT'S A REALLY DANGEROUS HACK BUT I DON'T KNOW WHAT'S WRONG" that were written ten years ago. I haven't even mentioned the legions of people attacking various parts of the internet for espionage and profit or because they're bored. Ever heard of 4chan? 4chan might destroy your life and business because they decided they didn't like you for an afternoon, and we don't even worry about 4chan because another nuke doesn't make that much difference in a nuclear winter.

On the internet, it's okay to say, "You know, this kind of works some of the time if you're using the right technology," and BAM! it's part of the internet now. Anybody with a couple of hundred dollars and a computer can snag a little bit of the internet and put up whatever awful chunks of hack code they want and then attach their little bit to a bunch of big bits and everything gets a little bit worse. Even the good coders don't bother to learn the arcane specifications outlined by the organizations people set up to implement some unicorns, so everybody spends half their time coping with the fact that nothing matches anything or makes any sense and might break at any time and we just try to cover it up and hope no one notices.

Here are the secret rules of the internet: five minutes after you open a web browser for the first time, a kid in Russia has your social security number. Did you sign up for something? A computer at the NSA now automatically tracks your physical location for the rest of your life. Sent an email? Your email address just went up on a billboard in Nigeria.

These things aren't true because we don't care and don't try to stop them, they're true because everything is broken because there's no good code and everybody's just trying to keep it running. That's your job if you work with the internet: hoping the last thing you wrote is good enough to survive for a few hours so you can eat dinner and catch a nap.

**We didn't start out crazy, we're being driven crazy**



Funny, right? No? How about this exchange:

"Is that called arrayReverse?"

"s/camel/\_/"

"Cool thanks."

Wasn't that guy helpful? With the camel? Doesn't that seem like an appropriate response? No? Good. You can still find Jesus. You have not yet spent so much of your life reading code that you begin to talk in it. The human brain isn't particularly good at basic logic and now there's a whole career in doing nothing but really, really complex logic. Vast chains of abstract conditions

and requirements have to be picked through to discover things like missing commas. Doing this all day leaves you in a state of mild aphasia as you look at people's faces while they're speaking and you don't know they've finished because there's no semicolon. You immerse yourself in a world of total meaninglessness where all that matters is a little series of numbers went into a giant labyrinth of symbols and a different series of numbers or a picture of a kitten came out the other end.

The destructive impact on the brain is demonstrated by the programming languages people write. This is a program:



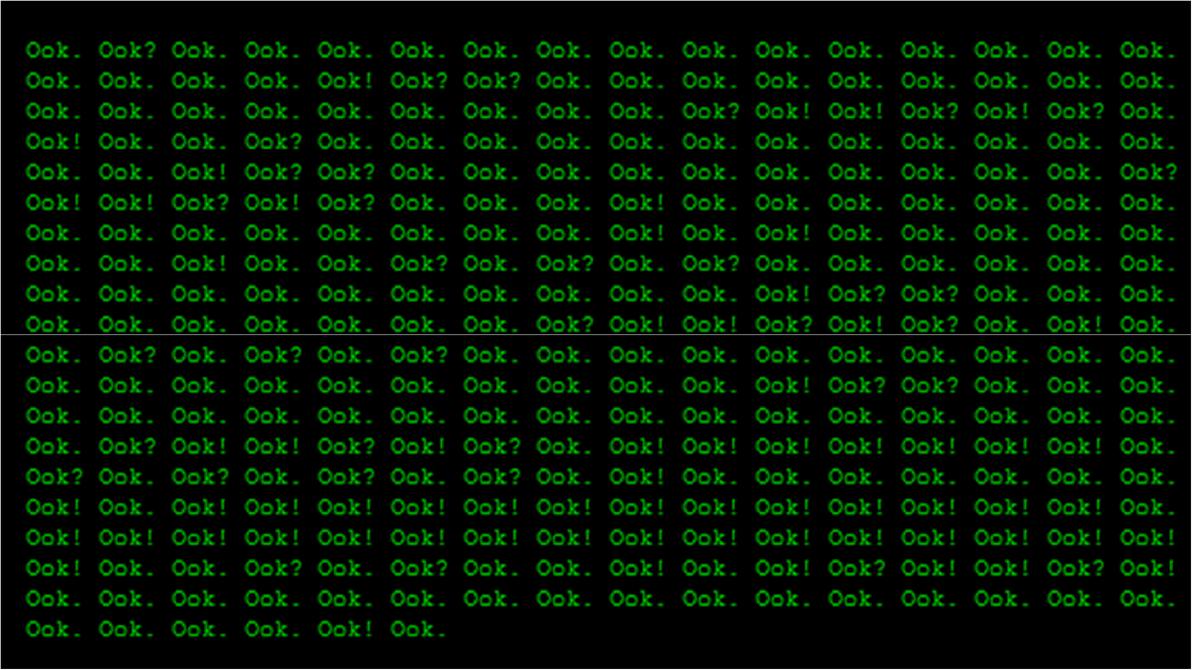
That program does exactly the same thing as this program:



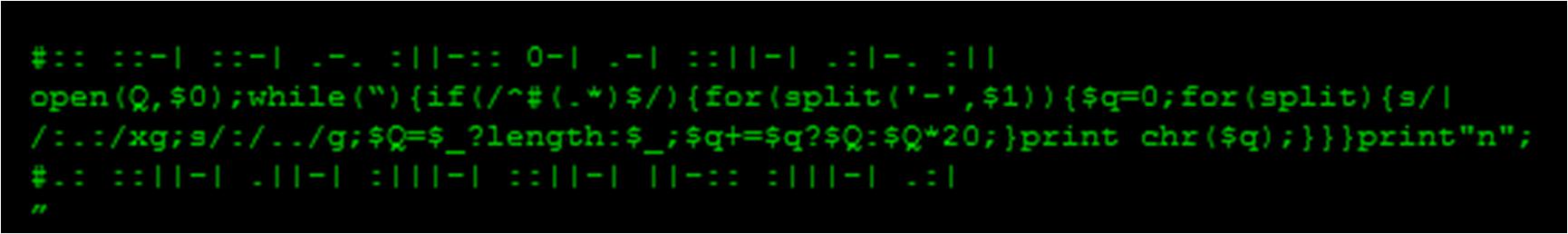
And this program:



And this one:



And once somebody wrote a programming language that let somebody else write this:



According to the author, that program is "two lines of code that parse two lines of embedded comments in the code to read the Mayan numbers representing the individual ASCII characters that make up the magazine title, rendered in 90-degree rotated ASCII art."

That program won a contest, because of course it did. Do you want to live in a world like this? No. This is a world of where you can smoke a pack a day and nobody even questions it. "Of course he smokes a pack a day, who wouldn't?" Eventually every programmer wakes up and before they're fully conscious they see their whole world and every relationship in it as chunks of code, and they trade stories about it as if sleepiness triggering acid trips is a normal thing that happens to people. This is a world where people eschew sex to write a programming language for orangutans. All programmers are forcing their brains to do things brains were never meant to do in a situation they can never make better, ten to fifteen hours a day, five to seven days a week, and every one of them is slowly going mad.

</rant>

So no, I'm not required to be able to lift objects weighing up to fifty pounds. I traded that for the opportunity to trim Satan's pubic hair while he dines out of my open skull so a few bits of the internet will continue to work for a few more days.

Sources :

https://www.stilldrinking.org/programming-sucks

https://www.laserfiche.com/simplicity/shut-up-imposter-syndrome-i-can-too-program/