O(1)

```
01010100 01101000 01101001 01101110 01101011 00100000 01000100 01101001
01100110 01100110 01100101 01110010 01100101 01101110 01110100 00101110

00000001
```

To start, you sit down with a cup full of Red Bull (you're scared you'll fall asleep). Flipping open your Mac, you hear the iconic chime and the machine whirs to life. After a while, the machine opens its eyes and greets you with the familiar *please enter your password*. With the Mac fully awake now, you can begin your process. Most importantly, is the music, so you open up the music app and put on some AC/DC. You know you shouldn't listen to music this loud at 11 p.m., but you also should be asleep, so you don't really care.

Next, you open up your preferred text editor, Fleet. It's too complicated for you (even with years of programming experience you still don't know what a Docker Container does) but you think it looks cool, so you still use it. You open up your project folder, `minigrep`, and stare at the code from the other night. At first, you wonder what you were thinking then, like why you added a comment that just said "help."

You know you are self-critical and anyone else would miss that comment if they looked at your code, but that doesn't stop you. Even though you need to finish the project, you can't get over the mistakes you made previously. You go through every line of code and meticulously check each character. You find extra spaces, and mis-indentations, nothing that affects the execution of your code. But you still can't just let them be there. Some deep instinct is telling you to spend all your time searching for mistakes, blocking you from finishing the project. When you get to the end of the file, you wish you hadn't. You wish there was more to go through. More mistakes to find. More code to be self-critical of.

`00000010`

What's written at the start is a message in binary, the language of computers. Computers don't have the millions of years of evolution that's gone into the human brain, so they use transistors. A transistor is the simplest unit of a computer possible, either on, 1, or off, 0. These units are called bits (short for binary digit). Everything else is built from these two states. Combine a string of eight bits together (resulting in 256 different combinations), and you get more than enough room for the English alphabet. If you combine 2,073,600 of these bits together, then an HD image is created. If you combine many billions of bits, then you've created a neural network, capable of understanding human speech.

You don't need some fancy computer science degree or a thousand-dollar machine to program. It just requires some creativity, patience, and a little extra time. You can use any machine and program in any language. All of this does not mean coding is easy, it's not. Usually, things don't work, and you have no idea why. But to some people, that's the appeal of it. The struggle to find the solution. The reward for when you finally solve the problem, and you get the `compiled sucessfully` notification.

The part that's hardest for me, however, is perfection. I sometimes go over every line of code again, searching for extra spaces or poorly formatted code, neither of which affect my code running. When I open up my computer, a small voice comes on in my head that tells me that I made a mistake somewhere. I can't stop until I find that mistake. But there's someone who's trained me to ignore this voice: my dad.

`00000011`

It's true that most people don't understand how computers work, but there is a small subset of the population that does: computer programmers. My dad is one of those people. He spends his time buried in the terminal (the interface between the human and computer), with hundreds of lines of green and yellow text flying by. One might be: `LinkedList is a raw type. References to generic type LinkedList<E> should be parameterized`. This

probably means nothing to most people, but my dad would understand instantly. He just forgot to add the object type to his list declaration.

My dad's worked from home his entire life. His office was connected to our mudroom, and I used to wander up there frequently. As a child, I loved to stand behind him as he lost himself in the world of the terminal. I watched the messages roll down his screen like rain dripping down a car window. Every once in a while, a message would roll down in bright red characters, and nothing else would follow. At the bottom of the message would be an indication of a `fatal error`. To me, `fatal error` sounded like a scary thing, but I never saw my dad frightened. He just typed away on his keyboard, and then magically the message was gone.

As a kid, full of curiosity and with much to learn about the world, this seemed impossible. How did the computer know what my dad was typing? How did my dad know that the computer knew what he meant? I used to think there was somebody on the other end of this "terminal" that was interpreting what he was writing and performing all the calculations.

Now, I know that's not how computers work (although it would be cool if there was a miniature city within them to run everything). Despite this, the curiosity I used to have is still there. When the code runs, I feel that same rush I felt years ago standing behind my dad.

```
00000100
```

The first time I ever touched a piece of code was in 6th grade when my dad gave me a subscription to a coding website. I used it for a couple of months, but then adolescence took full effect and I stopped. I instead focused on soccer, academics, and spending time with friends. I desperately wanted to be normal, to "fit in". I wanted to sit at the lunch table with everyone else and play stupid games after school.

That happened for two weeks at the end of 8th grade. I went to England on a school trip with my two best friends from middle school, their girlfriends, and 10 other kids who I didn't really know. I made a ton of British friends, had a lot of fun, and broke a few too many rules along the way. I was some ultra-outgoing version of myself, talking every chance I had.

I'm not entirely sure when it happened exactly, but I remember talking to my childhood best friend from back home, Max. We were catching up, and he asked a simple question, "Do you think you've become a better person?" It threw me off. For the first time in a while, I felt like I was becoming worse. I was having a lot of fun, don't get me wrong, but it wasn't helping me. I wasn't learning anything, and I definitely wasn't acting nicer that I usually was.

That night I sat in the cramped hotel room, with two other kids, contemplating my life. As with any deep contemplation, I had my AirPods in, probably listening to Manchester Orchestra (I don't remember the specific song, but it was most likely "I Know How to Speak"). There was something I felt that night that I had never felt before. I didn't know what to call it at the time, but now I know that it was the beginning of perfectionism. It felt like my life wasn't good enough, like I wasn't doing enough or learning enough. I don't remember what happened the rest of that night. I probably blocked it out, and archived it away in some folder somewhere. I do remember leaving that hotel room with a need to find my purpose. I was fourteen years old, trying to figure out my life trajectory.

00000101

Things didn't work out so well after that. My girlfriend at the time broke up with me, my best friend and his ex got back together, so he stopped talking to me, and I struggled to reconnect back with Max. I lost interest in school, zoning out during classes and barely putting in an effort in homework. I felt stuck, knowing I had a purpose somewhere, but without any idea how to find it. I spent all my time in my head, trying to figure out what exactly I was searching for. I stopped talking to most people, and became one of the quietest kids at school. After a couple months, I started to think that maybe I wasn't supposed to have a purpose. Perhaps I would spend the rest of my life like this. I was beginning to think I would spend my whole life feeling stuck.

Later that month, however, my dad and I went shopping for a new laptop for high school. We were trying to figure out what specs were appropriate for my use, and originally, we settled on a base model MacBook Air. He offhandedly mentioned that I would need more power when I

wanted to code. He said, “Well, if you want to learn to code, you should get a MacBook Pro.” We didn’t end up buying a computer, but that stuck with me. He thought I could program. I had only seen him code with his behemoth 15-inch MacBook Pro and Cornell Engineering degree behind him. What could I, a 14-year-old, do with a computer?

The following morning, I decided to find out. I snuck up to his office and plugged his Mac into his giant monitor. The screen began to glow, and I found the *App Development with Swift* book, an intro Swift book designed by Apple. Before I knew it, I was on chapter three, and it was nearly lunchtime. I had lost all sense of time, just glued to the text on the screen and the chime when my program ran successfully. That was how it went for many weeks, as I completed chapter after chapter of the book. Every morning I would set an alarm for 7am (before he would wake up), go up to his office, and code. He would eventually come up and have to kick me out, but by that time I had been coding for so long I saw lines of text when I closed my eyes.

```
00000110
```

I spent those months trying so hard to figure out who I wanted to be, that I forgot who I was. I was the kid who used to sort Legos by colors and counted all the goldfish before I ate them. I was the kid who asked just asked, “why” over and over again. Not to annoy the adult, but to get the reasons for why the things were.

My dad saw all of this. I’m uncertain if he consciously knew what was going on, but he somehow knew that I needed a hand. He knew that when I was left on my own, this curiosity and logical process lead me to perfectionism. I tried to get every Lego into their pile, and got frustrated when the piles never ended.

When my dad was there, it was as if he was holding my hand. He helped me focus more on the process of thinking. When I strayed too far, and got hung up on the result, like a wrong answer in homework, he reminded me that’s not what it's about. It’s about the process, going from a problem to solution. I was learning to solve problems. That’s what I think my dad has

been trying to show me this whole time. It doesn't matter whether I get the right answer or a perfect 100 on a calculus test. It matters that my thought process is improving.

I've struggled lately without his helping hand. College applications, AP Physics, and just the stress of being a teenager has been overwhelming. Although they're not on the other side of the world, my parents are not with me at Exeter. I'm usually too busy figuring out the net torque of a pulley is to call them. I've been hung up on the result rather than the process. Hung up on getting that perfect test, instead of focusing on actually learning the material.

00000111

To escape the pressures of life, I sit down with a computer. When I start, I just have an idea in my head, and a blank computer screen in front of me. One of the reasons I love coding so much is because it's a process. I need to transfer the thoughts from my head into an executable program (executable is a computer science term used to describe a program that could be run by a computer).

The first step in this process is to determine the logic that goes into the program. Like, will I need a loop? Or what's the class diagram going to look like? All questions that need to be answered before I write a line of code. Or rather, should be answered. In reality, however, I am under a time crunch and skip writing the plan down and just go with what's in my head (my dad says this is true in the professional world too). Whether it's a physical plan, I still need to think of how to get the idea out of my head. It's like having a goal, but without any clue how you will get there. With enough practice (and enough failures), you'll have a mild idea of what you're supposed to do. The next step is to make something tangible, to transfer the thoughts in your head to code on a computer. I could go on about the syntactic sugar of Python or how much I hate coding in Javascript, but that wouldn't make sense to most people.

Instead, I'll focus on perfection, the thing I've been chasing my whole life. Chasing the perfect grades, the perfect friends, the perfect personality. My parents always tell me perfection

is unachievable. My friends say, “It’s impossible.” I want to prove them wrong. Show them that, if I work hard enough, I’ll achieve perfection.

00001000

When I was writing this meditation, I struggled to transfer my thoughts into words on the page. I spent night after night writing and rewriting the sections. Night after night trying to make it good enough. The day before it was due, I conferenced with my English teacher. I told her that I was a perfectionist and I couldn’t figure out how to put what I was thinking into words. She responded, “I could tell.” Since then, I’ve rewritten this section six times, changed the formatting three, and changed the title twice.

This meditation will ever be good enough. I’ll never find the right words to say what I’m thinking. I think this is somewhat ironic. I’m so good and transferring my thoughts into code, yet can’t transfer them to words. Maybe that’s a sign of how I think. But I also try to find the appropriate code to do what I’m thinking. I chase that perfect code, perfect writing, like a computer stuck in an infinite loop.

The solution might be that I need to think more like a computer. Logic is the motivator for a computer, not emotions or feelings. The computer doesn’t care if I’m anxious or stressed. It doesn’t care if I’m coding at 11pm while drinking Red Bull and blasting AC/DC. I can spend all this time trying to make my code perfect, with the perfect format, the perfect runtime. But ultimately, the computer doesn’t know. The computer does not know the difference between perfection and non-perfection. If the code runs, it runs.

00001001

The quote at the beginning of this meditation is “Think Different”, written in binary. It’s a quote from an Apple ad campaign, back before Apple was a multi-trillion dollar company. I said before that most people don’t understand how computers work and I’ll say it again. There’s an even smaller subset of the population who are programmers that can teach others. These are the people, to overuse the cliché term, who “think different.” My dad is also one of those people. I

would be lying if I said he told me how to program. He didn't— he taught me how to love programming, to write the steps to solve an algorithm, to know when the algorithm is finished, even if it's not finished. Thank you for letting me stand behind you in your office all those years ago.

My dad once told me about O(1), a programming term (pronounced o of 1) that describes an algorithm which runs at constant time, no matter the size of input. It's the ideal runtime. One might even describe it as "perfection." He said it's easy to chase O(1), but it's usually physically impossible to achieve. There are some things that just cannot be done in a constant time, no matter how hard you try. I could spend my whole life trying to find a way to do it, but I would never be successful.

Part of me still wants to try. To spend my entire life chasing perfection, even if I know it's impossible. But what I'm beginning to realize, is that instead of focusing on that perfect O(1), I can focus on those little moments. The staying up late drinking Red Bull. The writing code for so long that when I close my eyes, I see the code. The midnight texts I send to my dad when I finally make it work and everything pays off. That's what programming is for me: using a computer to process my thoughts. Sometimes, that program won't be perfect, and that's okay. I'm not perfect either.