THE OPTIMIST

November 2024

Be Bold. Be kind.

Goings on: "...If everyone's trying to do a very difficult thing together and overshoots their limits, it becomes less of a tribulation and more fun," says RC's Rachel Vincent Petacat who has been at RC for over a decade. **Impossible Day** is a relatively recent RC tradition- It began in January 2023 and has run every 6 weeks since, meaning RC'ers doing a full batch will get two chances to participate. The tradition began as a way for folks to work on building the volitional muscles and over its recent history it's already produced beloved projects including a Bluetooth mesh network (Robin Neufeld and Amanda Galemmo), a game where piano notes are heard in loudness relative to their distance (Ryan Goldstein) and controller for a trombone game using an otamatone (Christina Tran). Impossible Day ran on November 19th, 2024, and will be back again in early 2025.

Ever experience writer's block or get suck in a loop of trying for perfection? Miriam Budayr, a current RC'er has. "As a classically trained musician, I find that sometimes focus on perfect so I try to do the exact opposite of making something perfect. I try to make it cringe, which makes me laugh and that makes me creative". **Wrong Answers Only Day** was a workshop held on November 15th. Around 20 people participated and created things in the wrong way to highlight mistakes. Some notable examples included resumes of failures and drawing iconic Disney princesses from memory. "Creativity is hard and showing up is hard, I hope this made it easier," says Miriam. When asked about doing it again Miriam felt optimistic. "I would love to see it happen again or have someone in Winter 2 take it on."

On November 8th, a group of around twenty gathered to explore physical making, electronics, and the Arduino for **Arduino Day**. Most had only a little familiarity with the tools but came curious and ready to learn. One attendee,

Justin T Schulz, went from not having worked with electronics before to being able to have a working metronome (a project he hopes to continue to work on). When asked about the importance of making things one of the day's organizers, Shenai Chen, said "Making something that you really really want to see in the world affirms that you are giving yourself permission to listen to yourself [... and] affirms that your needs are important. [It's also important] that you don't have to contort yourself to the shape of [proprietary os's] and yourself practice following through on a promise to yourself. [...] The more things you make, the more you understand who you are by seeing the finished artifact out there in the world!". A group of makers continue to experiment with making physical things so be on the lookout during your time at RC!

On November 19th Recurse opened its doors to alumni, friends, and curious programmers from across New York for **localhost** and a talk by Greg Sadetsky. The topic of the evening was Greg's recent project, Disco, and more broadly his love of websites and the open web. "Little websites, nothing bothering anybody, just existing. It's so beautiful" he waxed at the beginning of his talk. Greg's engaging speaking style had the audience rapt when he came to his pitch "If I'm pitching you anything it's this. You can put up a website on Disco. You can have a domain and everything, it's really cool." Curious RC'ers should check out the GitHub org letsdiscodev for more information.

"I'm nervous about having to take the subway, figuring out who to work with, and making the most of my batch" reported new RC'ers as they started their **first day of the winter 1 batch**. Gathered on Zoom on November 6th the batch began to get to know each other- sharing projects, ideas, hopes, and fears. "I'm so excited to be part of this community and work on new things. I hope I learn a new language" others shared. After the batch's first day, there was a one-day break from the US election before folks started to integrate into the hub and kick off in earnest. "I can't wait to see what happens!"

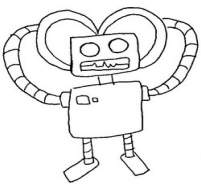
Welcome to The Optimist

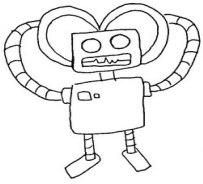
Welcome to "The Optimist". We're living through a time of profound technological transformation—AI is becoming an important part of our everyday workload, and changing how we interact with computers- sometimes in surprising ways. New programming languages are emerging and gaining traction, while web development seems to be in a constant state of reinvention. And yet, in all of this change, there is a distinctly human spark that shines and endures: our creativity, our curiosity, and our empathy for one another. We continue to find joy in the human act of making.

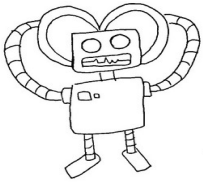
The Optimist is a snapshot of a moment in time, the start of Winter 2024 at the Recurse Center. We're celebrating the dreams in the algorithms and the people who make the machines run. Ahead, you'll find interviews with some of the minds at work here at The Recurse Center, news on happenings, illustrations, puzzles, and silly things. I hope that you'll pop your phone on its charger, take a break, pour yourself a coffee or tea, and settle in.

Be good to one another,

Nicole Watts
nicole.computer







This zine features contributions by **Lorenz Bauer**, **Sam Tetef**, **Luisa Vasquez** and **Be Birchall**.

The art throughout is by **Mia Abott.**

It was made in November 2024 at the Recurse Center- 397 Bridge Street in New York City.

Cake and Kale

Lorenz Bauer

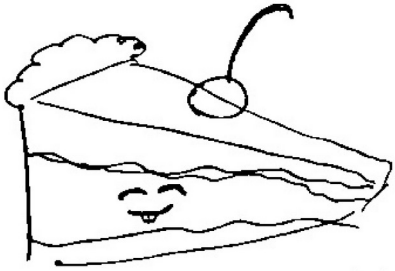
Much has been written about the merits of object disoriented vs. compositional styles of programming or the advantages and disadvantages of imperial over functioning languages. **Not enough, however, has been written on the subject of cake and kale** programming and how it relates to the experience of Recurse.

As practitioners in the field will recognise, there is a joy in a program well programmed or an executable well executed. That sugar rush to the head when a macro service first does a useful thing, when a bug has been zapped or a very hard thing has been overcome. Programming that feels like eating a whole cake in one sitting. As with any stimulation, we practitioners learn to crave this rush, so we begin to optimise. Use the tools we've honed over time, in the domain we're familiar with. The time to the next slice of cake shortens until all of our diet is shortening. Inevitable, this will leave one bloated yet empty. Bloated because the head is filled with arcane knowledge, empty because there isn't enough sustenance in a diet of flour, butter and icing. Like one's blood sugar, motivation will inevitably crash.

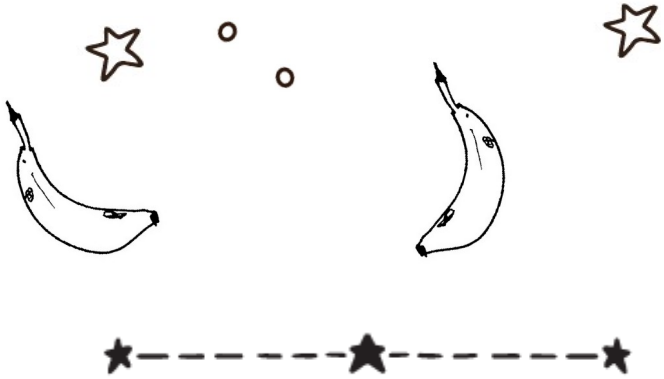
> “Consider snack-oriented programming: small projects, easy to digest, done on a whim”

So the natural inclination is to veer in the opposite direction. No more cake, we swear, while fixing up a bowl of maintainability microgreens with leet curds. From now on the learning will be hard but worth it, the equivalent of eating a head of kale a day. We eschew mutabal state, learn all algorithms, edit in ed. Through

this form of programming we will redeem our cake eating selves, become happier and healthier practitioners. Unfortunately, we can't have our kale and eat it too. Just like too much fibre will lead to constipation, a focus on kale programming may stop one from progressing. Inevitably progress is slow, motivation wanes and cake sounds delicious again.

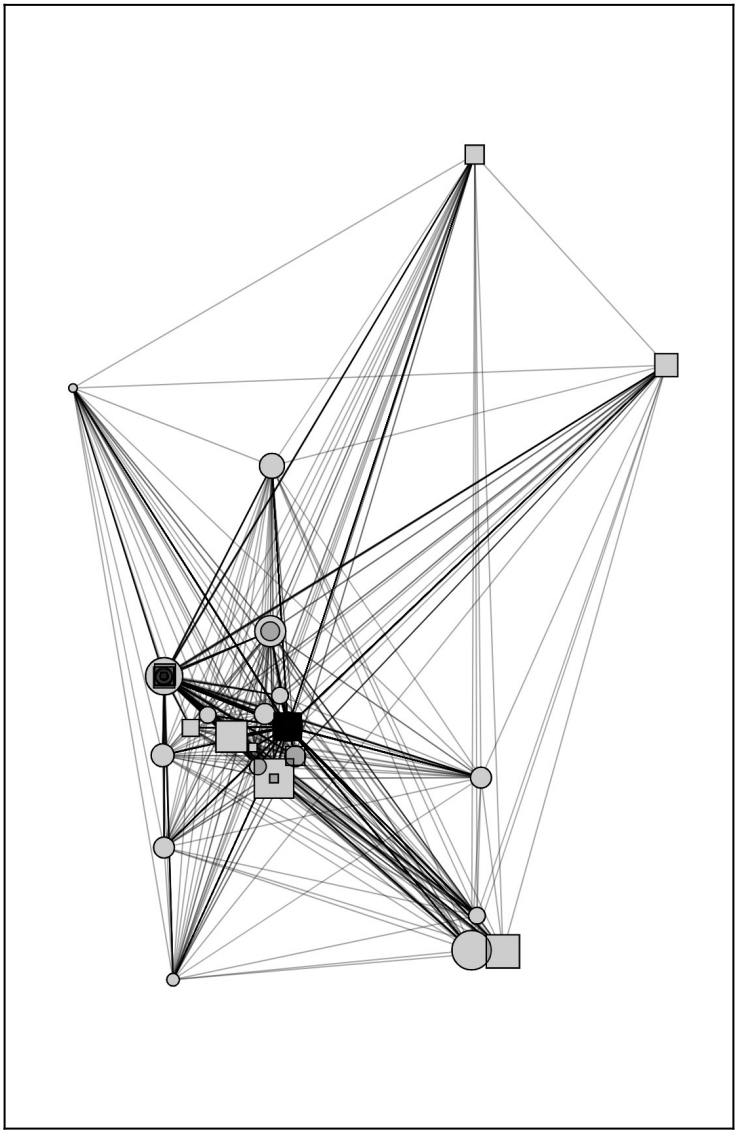


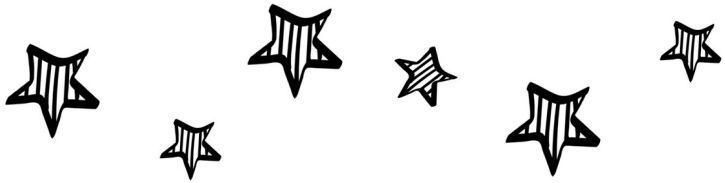
Here is where the danger lies. It is a false dichotomy to separate the world into cake and kale. As one may know, almonds are rich in fibre while chocolate "may [lower] the risk of heart disease and stroke, [support] cognitive function, and more". In other words, chocolate covered almonds are a superfood like kale while being a lot more delicious. The author therefore urges you to consider snack-oriented programming: small projects, easy to digest, done on a whim.



Interconnections

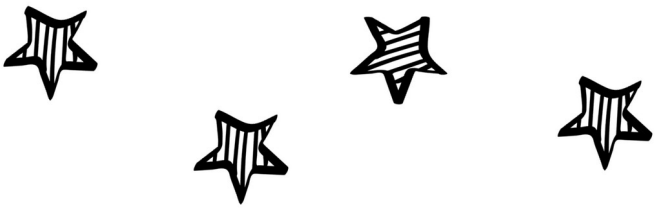
Sam Tetef & Luisa Vasquez





You're looking at a **map of the locations and interconnections of Recursers**. How did we map people? Well, first we sourced where people are from using the RC directory, and then we converted those locations into latitudes and longitudes that are mapped to the canvas. We used shapes to represent individuals: a circle (W1 24) or a square (F2 24). Each shape is scaled based on the number of characters in each person`s name, and all the shapes are partially transparent. The layered shapes then overlap in certain areas to highlight a density of Recursers in places like New York. Finally, we drew lines between all the individual shapes to show that even though we may be far geographically, we`re all connected through RC.

- Samantha Tetef and Luisa Vasquez



Edge Cases at Infinity

Be Birchall

"That's just an edge case," you might hear a programmer say, or "it works... except for edge cases." When I entered the world of software development, I was surprised to hear edge cases dismissed as afterthoughts.

In my previous life studying math and philosophy, edge cases were the clues you'd look to first. "What happens when it's zero? What about the empty set?" might be your first questions when you're trying to prove something. A proof that works most of the time isn't a proof at all. When trying to understand a concept, you replace your fuzzy idea with something precise, then test your definition with atypical cases.

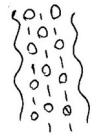
In software development, edge cases don't feel like clues to anything. We glide from atypical and hard to think of to uncommon and unimportant. If we have an atypical case that we decide is important, we can treat it separately.

Paradoxes are the ultimate edge cases. Ancient Greek philosophers discussed cases that are so atypical that they are beyond belief, and the word comes from the Greek roots para- (meaning beyond or alongside, as in "Paralympics" and "paragliding") and doxa for belief.

"This statement is false." If the quoted statement is true, then "this statement is false" is true, and so it is false. But if it is false, then it is saying something true! This is the liar paradox.

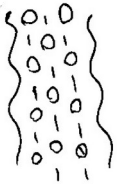
Let's look at an example testing weird cases against a fuzzy idea made precise. What does "same size" mean? Make it precise by saying two sets are the same size if you can map them one-to-one, so everything in each set is mapped to a unique thing in the other.

A weird case: is the infinite set of all counting numbers (1, 2, 3, ...) the same size as all points on a line between 0 and 1? Represent the numbers on a line between 0 and 1 in binary, so each is a sequence of 0's and 1's. Mapping them one-to-one with the counting numbers means putting them in an infinite list like this: $1 \rightarrow 0.11000$, $2 \rightarrow 0.00101$, ...



Consider the "diagonal" number d defined by looking at the first binary place of the first number in the list, the second place of the second number, and so on. The nth place of d is 1 if the nth place of the nth number is 0, and 1 otherwise. But then d differs in at least one place from each number, and so it cannot be in the list. This is Cantor's diagonal argument establishing different sizes of infinities.

Different sizes of infinity? Bertrand Russell thought there had to be a mistake. Trying to find a flaw in the proof, he instead discovered "a new and simpler contradiction." Applying the diagonal style of reasoning to the set of all sets, he asked whether the set of all sets that are not members of themselves is a member of itself. Either assumption leads to the opposite conclusion!



Frege was trying to derive math from logic. But his system allowed for the set of all sets, and so to contradiction. Russell looked to the ancient Greek philosophers for a clue. He thought his paradox felt like the liar paradox: both allow a "vicious circle" of self-reference. You can avoid vicious circles if statements and sets are in a hierarchy of "types," he thought, where each one has a type and can only reference those of a lower type in its definition.

Alonzo Church, some years later, was trying to make the fuzzy idea of computability precise. So he introduced lambda calculus, which he also wanted to use in another attempt at deriving math from logic. He added types to lambda calculus when he realized his first version was vulnerable to Russell's paradox.

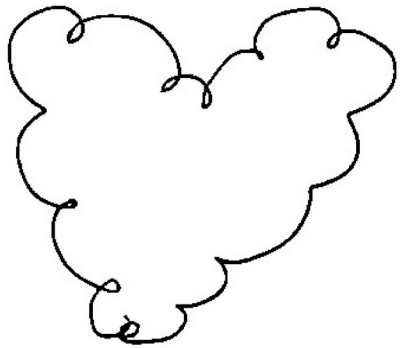
These were the early days of typed programming languages. ``How did Bertrand Russell get into web development?" I found myself thinking when I joined a team that was using TypeScript, the typed extension of JavaScript. Types help developers avoid bugs: by restricting the inputs functions allow, you make sure code fits together before you run it. That way, you don't have to think so much about edge cases. But we can view edge cases with less derision now that we've seen how paying attention to them led---via a tale of paradox and infinity---to the very techniques that help us avoid them!

Profiles

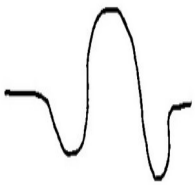
The Recurse center is filled a vibrant cast of wonderful humans. Many of these interviews were conducted in-person and the rest were over text. They have been editted for length and clarity.

Christina Tran is an RC'er and has been programming for 6 years years. **How could computers be different?** I wish Windows on the Desktop wasn't so overwhelming. As I work through "The Missing Semester," I'm also realizing that maybe we don't need GUIs for everything. I'm sm also realizing that maybe we don't need GUIs for everything. I'm so impressed when someone opens a terminal to do calculations rather than going to the calculator app. **Fave website?** neal.fun and neocities **Fave computer book?** Recently, *"The Nature of Code"* by Daniel Shiffman.ough, when I'm not at the computer I don't want to think about computers. **How does AI change programming?** I think AI will separate people who really like programming from people who ut right now companies are deciding how AI gets used. I'm not optimistic about it. I think we're going to see the definition of "what a programmer is" change. There will be more people programming but it will be different from what we do today. **First computer loved?** My family had a dinosaur of a computer that had a sticker with three birds. It wasn't really my computer but I would sneak onto it and run emulators. **Overrated / overused Emoji?** The crying-face one. It's used in so many contexts. People use it when something is funny, sad, or when they're happy.

Emily Bernier is a staff member at RC and has been programming since she had a graphing calculator- almost 25 years. **Journey to RC?** I heard about RC from a friend who had come when it was still called Hacker School. I applied, came to a batch, and then was looking for a job with a culture like RC's. They were hiring and I work here now. **Something about RC most people don't know?** There was an incident with a detective... We used to operate out of somebody's office. Maybe it was Etsy? [editor's note: RC did have space in Etsy in 2012] **Fave spot nearby?** Dekalb market. It's an underground street market and there's a big variety. They capture the street market feel while being inside instead of seeming like a mall cafeteria. **First computer loved?** My mom worked at a law office and they had word processors running word perfect and windows 3.1. I used to use them as a kid and they were magic. **The Hub is haunted?** I haven't heard that! I don't get the sense that it is but maybe you'd have to ask the church downstairs. **Overrated / overused Emoji?** I don't know. I think emoji are appropriately rated. They're great. How about an underrated emoji? It's the dancing blob. It's joy, enthusiasm, and general yay. The dancing blob captures the vibe we should all aspire to.

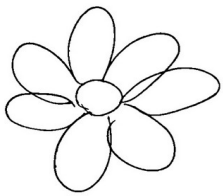


Finn Clarke is a staff member at RC and does not consider themselves a programmer. **Journey to RC?** I was working at a law firm and thought maybe "law school" but I didn't like it. I heard about RC from an email and they didn't want a traditional cover letter. I loved that. I started here earlier this year.
Something about RC most people don't know? Our corporate name is "Five Blades Inc." which was based on an Onion article about how razor companies kept adding more blades. Now five five-blade razors are common. **Fave spot nearby?** Bahn migos! I get the vegetarian Bahn Mi and a



bubble tea. Also a shoutout to the falafel cart on Court Street. **The Hub is haunted?** You know I do sometimes hear the clickety-clack of keys late at night but when I go look there's no one programming. I think it's a nerd's ghost though so I'm not scared. **First computer loved?** My aunt gave me a plasticky green clamshell Mac laptop. It barely worked and I loved it. **Overrated / overused Emoji?** The cool glasses emoji- I use it too much.

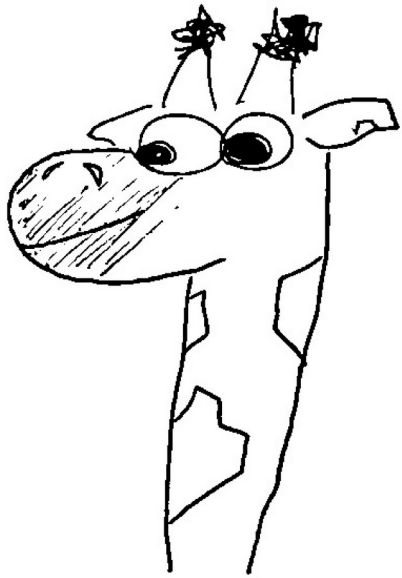
Jesse Chen is an RC'er and has been programming for 18 years. **How could computers be different?** I wish it were easier to dig into how tablets work. A lot of computers in the 80s and 90s (including Macs!) were very malleable. We've lost that ability with closed-platform tablets and phones. We gained security and ease of use but something important was lost. **Fave website?** ncase.me **Fave computer book?** *"Ra"* by qntm. It's an alternate history where magic becomes an engineering field. **How does AI change programming?** It'll for sure make it harder to do code reviews and reason about how code works. The next paradism after LLMs will have a lot of potential but it's more than 10 years out. I think it's important to remember that accountants had a strong reaction to spreadsheets and they still have a job. Trying to predict the future is like trying to predict that influencers would have been a thing in early 2000 but overall I'm optimistic. **First computer loved?** An IBM PC AT that my Dad bought at a computer fair. **Overrated / overused Emoji?** Cowboy hat



Michael Chung is an RC'er and has been programming for 14 years. **How could computers be different?** I think spatially with computers- I have a desktop just for personal things, another for work things, and another for church things. When I plug into an external monitor my windows get all mixed up. I know there are ways to fix this but more generally I think in categories and I wish my computer was better at maintaining that state. I need compartmentalization. **Fave website?** Functionally it's Notion. I

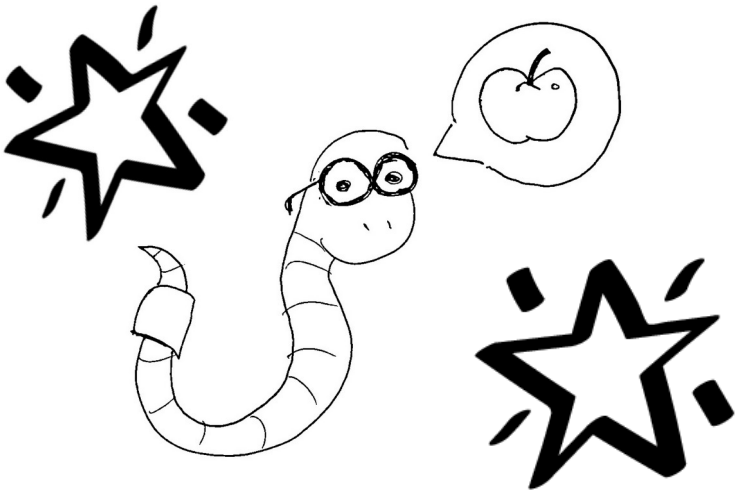
planned my wedding with it. **Fave computer book?** I'm enjoying *"Designing Data-Intensive Applications"* by Martin Kleppmann **How does AI change programming?** So far we've seen it take on the job of writing boilerplate job. I've heard from other programmers that it's making the task of understanding large codebases easier. I feel optimistic about AI in programming but I feel less optimistic about it in other domains and fields. I think in the future more people will be able to use programming. **First computer loved?** Gameboy Color playing Pokemon Gold. It was my first experience with a computer that was personal and mine. **Overrated / overused Emoji?** I hate the tongue-sticking-out emoji. Facebook used to have a different emoji set and it had bigger eyes and was a little sad. The current one is too silly. I use the upside-down emoji now to give the same feeling.

Nick Bergson-Shilcock is a staff member at RC and has been programming for 30 years (the Apple II on the fourth floor was his computer). **Journey to RC?** The super quick version is I always wanted to start a company. I met Dave and Sonali and we decided to start something. For a while, we solved problems that weren't really problems and then we stumbled upon the idea for Hacker School which became The Recurse Center. **Something about RC most people don't know?** Daniel Ek, the CEO of Spotify, visited during our second batch. He wasn't so well known at the time. He told us that the space we were working in had previously been a recording studio where Bob Markley had recorded. **Fave spot nearby?** Sweet green or Naya are gotos but my favorite spot is Sanpanino, an amazing



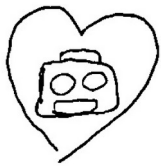
sandwhich spot. **First computer loved?** A Mac Plus. It had MacPaint on it which was mind-blowing at the time. **The Hub is haunted?** No, definitely not. Ghosts don't exist. **Overrated / overused Emoji?** Confetti. It's used so much and it's not that celebratory.

Niko Lazaris is an RC'er and has been programming for 8 years. **How could computers be different?** I wish there were less screens. We need more buttons. It sucks that everything is just a black rectangle. **Fave website?** The Internet Archive **Fave computer book?** *"Coders at work"* by Peter Seibel **How does AI change programming?** Slowly programming is going to become more creator-directed and high level. We're in an AI hype bubble right now. I think AI is going to lower the barrier to creating things- for example, I had a neighbor who was able to make a game and when I talked to them I was excited to talk about data structures. I asked them how the data was being stored and accessed and it turns out they are using a LLM. The data structures didn't matter. Still, I think the more people building things, the better. **First computer loved?** I can't exactly remember but I know that it was running Windows 95. **Overrated / overused Emoji?** It's the face ones. We can do better in 2024.



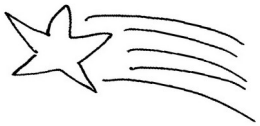
River is an RC'er and has been programming for 17 years. **How could computers be different?** I wish that I could write a program that graphics and sounds without having to be an expert in complicated libraries. I also wish there was a well-defined instruction set for graphics cards but there are economic reasons that there isn't. But, one of the deepest things for me is that I wish it were easier to share code between languages. It works if the "other language" is C but if I write F* and you write Ruby it's really hard to make the code interoperate. It means that starting a new programming language isn't just a technical challenge, it's a marketing one. **Fave website?** Recurse's Doorbot- it calls me the elevator. **Fave computer book?** *"Practical Foundations for Programming Languages"* by Robert Harper. **How does AI change programming?** I'm not particularly educated on that issue but I think there will probably be more programmers in the future. **First computer loved?** In middle school my parents got me an ugly beige computer, it probably ran something after Windows 95. **Overrated / overused Emoji?** The Thinking face one. I feel like it's mildly derogatory.

Saleh Alghusson is an RC'er and has been programming for 8 years. **How could computers be different?** I wish they were easier to use. Computers [and software] are built by computer people and for computer people. For example, I have a neighbor who I help. I noticed that a browser is hard for her. Every website has a different layout and works differently. When she needed to log in she opened a Word document because, for her, the accessibility of being able to find her password is more important the security. Adding a password manager would have meant adding more complexity. **Fave website?** Neal.fun **Fave computer book?** *"The Art of Computer Programming"* by Donald Knuth. It made me love programming. **How does AI change programming?** I think developing software is going to get a lot easier. I also noticed that lots of software isn't designed for AI. I was signing up for x.com recently and it took me 10 minutes to figure out



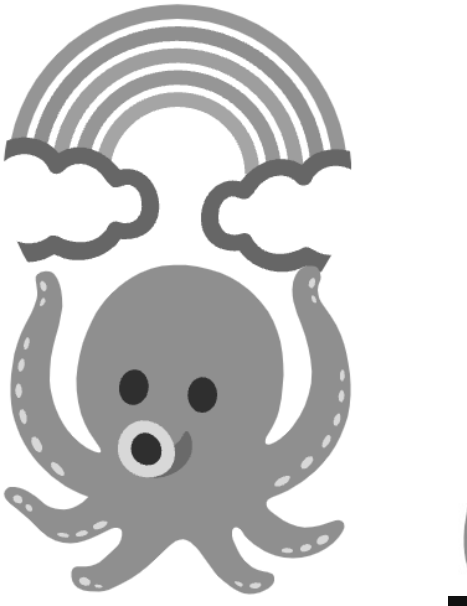
the captcha- they're getting too hard because they're designed to defend against AI. I think in the future we're going to have a lot fewer programmers and I'm not optimistic about it because AI abstracts out all of the steps of writing a program. **First computer loved?** I switched from a Windows computer to a Macbook Air and I was amazed at how pleasant it was to use. It changed how I thought about what computers could be. **Overrated / overused Emoji?** The water splash one- what does it mean?

Samantha Tetef is an RC'er and has been programming for almost 10 years. **How could computers be different?** Infinite battery life, waterproof, doesn't hurt your eyeballs as much, and all of them have touchscreen. **Fave website?** xkcd.com **Fave computer book?** Grace Kwak (a current RC'er) has a book called *"The Glitch: A Novel"* that I can't wait to read! **How does AI change programming?** I imagine AI will give people a little boost at the start of learning curves, especially for learning new languages, but it will still take strong fundamentals to maintain that momentum to get the rest of the way there. **First computer loved?** I had a Lenovo Yoga that got me through most of college and grad school (the touchscreen was super helpful during Covid because I could continue taking hand-written notes and write digital homework solutions). **Overrated / overused Emoji?** :smile: (I don't know, I like them all, but the plain smile is a 'lil boring?)



Zab is an RC'er and has been programming for 3 years. **How could computers be different?** Every once in a while we should just start from scratch. We've just kept adding on top of old things and we need to start with new ideas. Command line interfaces feel whack given where we are. **Fave website?** The Internet Archive **Fave computer book?**

"Neuromancer" by William Gibson **How does AI change programming?** AI is going to do it all. We'll be in that solarpunk future we've all imagined. Everyone will be making things. Think about the jump from assembly to Python. AI is that next thing. It's the next set of abstractions. Am I excited about AI? Yeah, I am. But optimistic? TBD. **First computer you loved?** On my first computer, the family computer, there was a cool gimmick where the icons were animated. The trash bin was like a dinosaur that threw things. It was a small detail but it made the computer come alive. **Overrated / overused Emoji?** The Poop emoji. I don't get it but people love it.



John Hergenroeder (m7'21) created emoji.octopus.holdings to celebrate the Zulip emoji line-height bug that led in part to it becoming RC's unofficial mascot. It intentionally has a lot of easter eggs – Jesse Chen

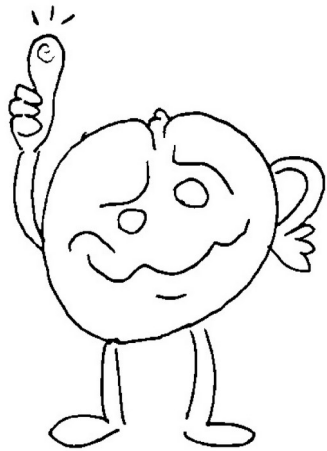
The Outlook

Trust the *random numbers*, here's what's ahead for you in your horoscope

January & February Your system is experiencing performance bottlenecks. This is the perfect moment to invest in personal infrastructure. Consider learning a new programming language, taking an online course, or upgrading your mental toolkit. The lag you're experiencing isn't a bug—it's a feature signaling growth. Whether it's professional development, a new hobby, or expanding your network, now's the time to allocate more memory to your personal development. Don't just debug; optimize.

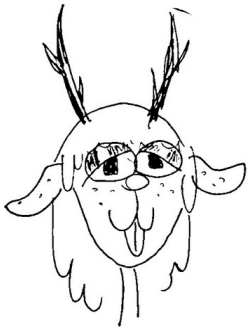
March & April Warning: Mental stack overflow imminent. Your processing capabilities are near capacity, and continued input without proper memory management will lead to system shutdown. Implement an immediate self-care interrupt. Meditation, exercise, or simply disconnecting from digital stimuli can help you clear cache and prevent a total system meltdown. Remember: Even the most robust servers need routine maintenance.

May & June Current operational status: Chaotic. You're experiencing what programmers might call a 'stochastic event'—unpredictable and seemingly random interactions that defy standard algorithms. Stay adaptable. Your current environment is running on high-entropy mode, but stability is on the horizon. Treat these weeks like a complex debugging session: observe patterns, remain patient, and trust that order will emerge from the apparent randomness.



July & August You're currently in the final stages of a complex problem-solving sequence. The solution is tantalizingly close—like a compilation that's 99% complete. Resist the urge to cancel the process or start a new project. Your persistent debugging is about to pay off. Whether it's a personal challenge, professional project, or internal conflict, your determination will successfully compile your desired outcome

September & October Network connection: Strong. Your interpersonal bandwidth is wide open, and communication protocols are optimized. Engage in conversations, attend networking events, or join collaborative projects. These interactions aren't just social—they're potential innovation pathways. Like open-source collaboration, the most groundbreaking ideas often emerge through unexpected connections and shared intellectual resources.



November & December You're operating on a different operational system from those around you, and that's not a vulnerability—it's a feature. Your unique perspective, unconventional approach, and distinctive problem-solving methods set you apart. While others might not immediately comprehend your workflow, your efficiency and adaptability are your strengths. Embrace your distinct architecture. True innovation has never been about conformity, but about bold, unapologetic differentiation.

After RC: An interview with Sonali Sridhar

"I think there's a way that non-tech people undervalue the importance of relationships. Not to romanticize too much, but in the early 2000s things were different - people were building the web and connecting things because it was cool and an interesting problem." Talking with Sonali, it's hard not to feel excited about the work that technical people do. She's been at RC since the very beginning - more than 13 years and has had the chance to work with many notable tech people. "There was a time period when we had a lot of folks who worked on programming languages - the creators of Elm, Julia, and ClojureScript were part of RC at different times. It's interesting how we go through phases." In the beginning, she tells me, there was a focus on making New York a serious tech scene, and the focus of early batches reflected that. In the early days, RC used to play a more active role in people's projects, and staff members participated in batches more actively as contributors. But over time, RC changed its structure, allowing RC'ers to explore more freely. "A surprising thing happened with that change - we got more fun projects but we didn't lose the focus on serious programming."

The topic of my conversation with Sonali is "Life after RC," and we can't help but begin with the changes in recent years. "Things really changed in 2022," she tells me. "It's a whole cascade of things - it was harder for companies to go public, so it was harder for that capital to fund new ventures, and because of that there was less turnover in engineering roles, and it all plays into each other." But she lights up when she tells me how excited companies are to hire people from RC. "The reason most companies hire from RC is because they are so impressed with the culture. They want to bring some of that into their company."

With American Thanksgiving next week, my thoughts are on gratitude, and I ask Sonali what responsibilities we have to RC after we

become alumni. "First of all, I think alumni is an interesting word - we don't consider someone an alum until they're one year out from their initial batch. We think of ourselves as partners to RC'ers through their batch and finding a job." And as to what responsibilities we have, she's quick to point out, "None. You don't owe RC anything, but we are delighted when RC'ers become part of our extended network, when they bring back ideas, give workshops, and consider hiring through RC for future roles."

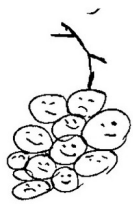
"We don't consider someone an alum until they're one year out from their initial batch. We think of ourselves as partners to RC'ers through their batch and finding a job"

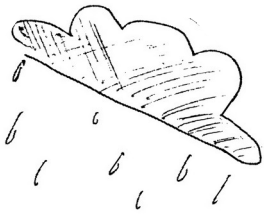
I asked her about what most people do after RC. "It's a mix. Most people look for a job, and we're able to help many of them. For some, that's direct placement. For others, it's guidance on resumes, interviewing, and getting through the process. Sometimes we have open roles but haven't found the right RC'er yet, and sometimes we have an RC'er but don't have the right role for them yet. The placement really matters, and we want to honor people's preferences in a job - be that location, industry, or other factors. For some people, though, they need a break. They take a break and travel. Others go back to a job because they were on sabbatical, and for a small few, they're retired and don't need to work."

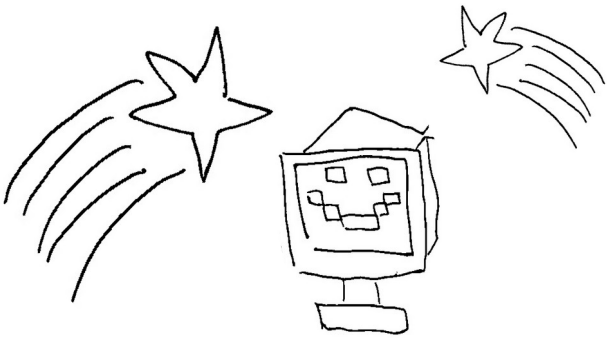
As our interview closes, the topic comes to emoji. "What do you wish people were using more?" I ask her. "Ooo, the hammer and wrench! I love that one."

Notes

Algorithms, doodles, love letters, and UI sketches







Be good to one another.