I was about 10 years old when I first picked up one of my father’s old books, *The C Programming Language*, and began learning to program. I started with simple exercises before moving on to building choose-your-own-adventure and graphic-based games. Soon I was searching online for other programming languages, creating my first website and blog when I was 11. My school didn’t offer programming classes, so I wasn’t sure if programming would have any further use in my academic career, but I knew it was a stimulating activity that I absolutely loved.

I heard about hackathons—24-hour programming sprints—through online web development communities when I was 13. It took me another year or so to actually sign up for one, but in the meantime, I continued honing my skills.

**A Different Kind of Hacking**

Hackathons focus on making something new or devising an innovative use for something that already exists. Usually, that means creating something interesting or useful from existing software—a virtual reality (VR) application that converts sign language to speech, for example, or a monitor that alerts users to the availability of open bathroom stalls at a crowded venue.

While hackathons generally promote collaboration over competition, it is exciting to race against the clock to finish projects and to compete against other hackers to develop the most innovative and technically challenging applications. The best projects sometimes win prizes and glory, but the true reward is being able to complete your project and walk out with new skills under your belt.

Most hackathons are sponsored by technology companies that provide free merchandise in exchange for hackers utilizing their software or interviewing for jobs with their companies. There’s usually an unlimited supply of food and coffee, and, to break up the hours of coding, fun events like karaoke and cup stacking to help hackers relieve stress.

Hackathons also provide access to a community of mentors—experienced hackers and developers who attend for the purpose of helping out less experienced programmers. They might also run workshops, demonstrating skills they’ve acquired and sharing their projects with interested hackers.

**First-Time Jitters**

My first hackathon was Pearl Hacks in 2015, held at the University of North Carolina. I applied after hearing that high school students could attend, and was pleasantly surprised to find out several weeks later that I was accepted. Then, one Friday in March, I packed my bags and my laptop, met the bus the hackathon organizers had sent to NYU, and rode off into the evening, awaiting my adventure.

Until the bus made its next stop, at Rutgers, I was the only high school student on the bus. But I made friends fairly easily with the college students, many of whom shared their past hacking experiences and gave me tips on how to survive my first hackathon: take regular breaks, sleep, and don’t stress too much over bugs.

By the time we arrived at UNC, I had already formed a team with some of the other high school students and began brainstorming to help us come up with ideas for our project, we went to work. Hackathons occasionally have themed prizes that encourage developers to create apps or games that fulfill a certain purpose. For Pearl Hacks, one of the themes was promoting sexual health.

My teammates and I decided to build a game that would demonstrate aspects of sexual health. We worked on a game development platform for iOS, eventually creating an infinite runner (a game in which the player continuously moves forward through the game world) that demonstrated the importance of contraceptives in a playful way. Our app ended up winning the Best High School Hack category, along with a trip to Google headquarters.

**The Aftermath**

Over the next few months, I attended six more hackathons. At every one, I learned something new, whether it was making VR apps, programming for Android devices, utilizing new APIs, creating web applications, or other skills. I also found out about hackathon-oriented Facebook groups and other online communities, including Major League Hacking, a company that helps college and high school students organize events at their schools.

I wondered why my school didn’t have its own hackathon. At the time, high school hackathons were not that common, but I figured it couldn’t hurt to try. So, in May 2015, with the help of a few of my friends—Sebastian Cain, Kenneth Li, and Yasmeen Roumie—I started outlining plans for what would become StuyHacks.

Organizing a hackathon was one of the most difficult tasks I’ve ever undertaken. From juggling outreach, sponsorship, and venue searches to talking to administrators—all in addition to schoolwork—I was exhausted. Nonetheless, it was exhilarating to see StuyHacks evolve from an idea into a full-fledged event. Dozens of college
students registered to mentor, and hundreds of high school students from as far away as Pennsylvania signed up to attend. As the days ticked by, my enthusiasm and excitement gained momentum.

Bring On the Hacks
The first StuyHacks began on a Saturday morning in October 2015. I arrived early to the venue, a co-working space in Midtown Manhattan, with boxes of supplies and food for attendees. Piles of prizes, merchandise from sponsors, sound equipment and laptops, and other paraphernalia filled the heavy crates to the brim. My teammates soon arrived with their own supplies and helped prepare the space. We hung signs with wifi passwords, laid out the food and information, and set up tables for registration.

We checked in the hackers as they arrived and then spent the rest of the day running around, helping out newer coders, setting up projectors, announcing workshops, bringing out food, or ushering mentors and sponsors around. I glimpsed a range of projects—a VR medical emergency simulator, a Javascript math game, a one-handed calculator app, an iOS DJ mixer, a remote-controlled car, and others. Seeing the attendees’ incredible work and their enthusiasm was inspiring and gratifying.

The day flew by, and soon I was announcing the winners for best design, best overall, best technical, best game, and best first-time hacker. All the long nights we’d spent preparing every little detail for StuyHacks had helped it run more smoothly than I could have even imagined. As the attendees filed out and we started to clean up the space, my team and I were already thinking about our next hackathon.

Community Relations
Since then, I’ve become even more involved within the hackathon community. Whether it’s attending events across the country on weekends, working on projects with new friends, discussing trends in technology and hardware in online groups, or mentoring younger hackers at events like CodeDay (coding sprints geared toward education), these weekends have changed the way I see myself and my role in the computer science community.

Many people think computer science is a solitary pursuit, but hackathons have shown me that it’s a collaborative effort. Whether you begin as an expert in a language or a complete newbie, there are always things to learn, people to meet, and new technology and skills to discover. If you’re new to programming or interested in learning, consider attending a hackathon. Whether you win a prize, make a friend, or develop an app you’ve been meaning to build for a while, hackathons provide the perfect atmosphere to bring your ideas to life.

Sharon Lin is a senior at Stuyvesant High School in New York City, where she runs StuyHacks and BitxBit Camp. She is a coach for Major League Hacking, as well as a finalist for the Google Science Fair and the Thiel Fellowship. She currently works on the Android development team for the U.S. Department of State and Facebook. She was named a White House Champion of Change, and has appeared on Voice of America and the Qualcomm #WhyWait Invent-Off.