

# CS First

*unplugged* 

Computer Science activities without a computer

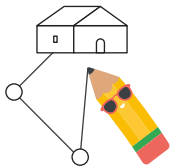


## Computers connect us

Computer Science helps people in many ways in their everyday lives. Internet and phone apps let people chat and find others who share their interests, even when they can't be together in person. These tools were built by computer scientists.

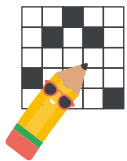
Try these three activities that show how computers can connect people. You won't even need a computer.

### 1. Network a neighborhood



Some people use map programs every day. Finding the best path from one place to another is a tricky problem that can be solved with the help of Computer Science! Try to connect all of the buildings in a town. Use as few spaces as possible.

### 2. Encode an emoji



Computers can't "see" pictures. Images need to be turned into numbers, a process called encoding, for a computer to recognize them. Help a computer recognize emojis by encoding them.

### 3. Send a secret message



When people chat on the Internet, they need some messages to stay private. Learn a trick that's been used for thousands of years to keep secrets. Then use it to send a message to a friend!

## CS Career Connections



**Peta-Gay Clarke**

(she / her)

**Community Connector**

**Hometown:** New York City, NY

**How she got her start:** Peta-Gay Clarke studied Computer Programming in college and then worked in many tech-related roles.

**What she does at Google:** Peta teaches CS at Pace University. She also leads Code Next, a program that helps Black and Latinx high schoolers become tech leaders.

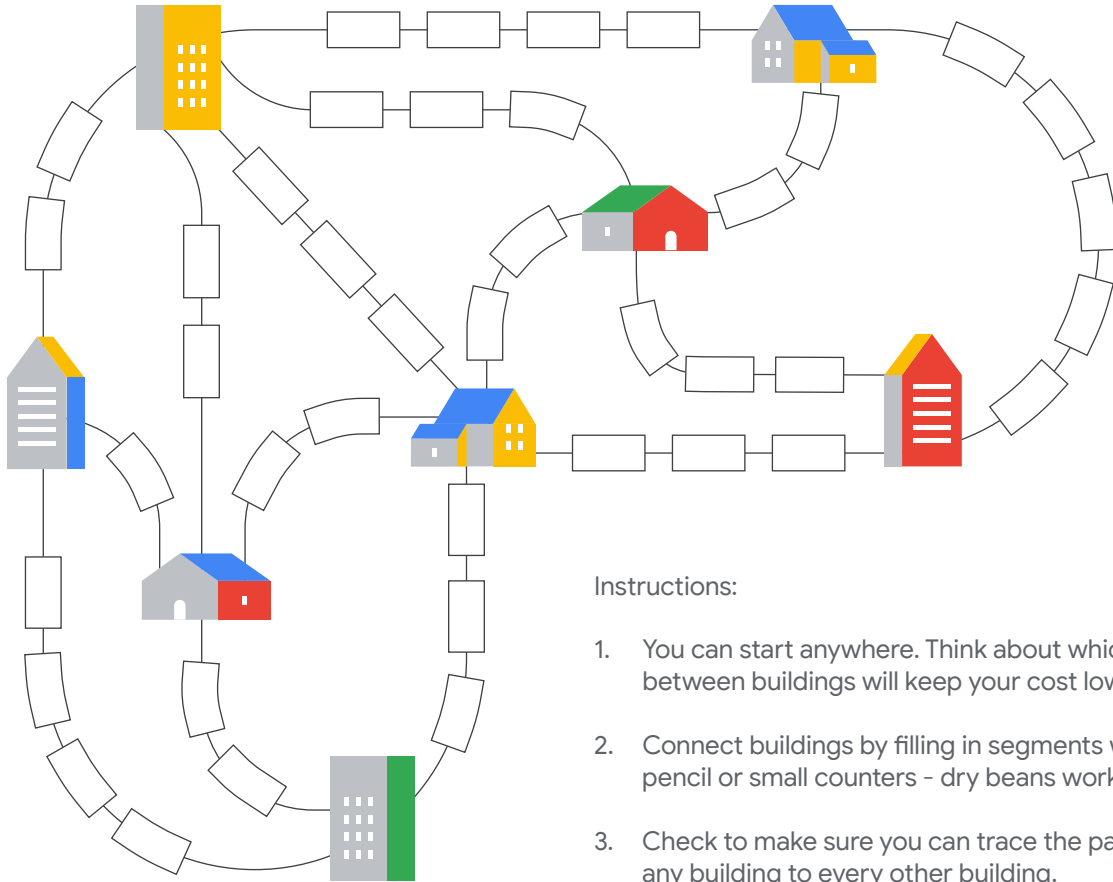
**Why she loves her job:** Peta is a first generation Jamaican-American. In her job Peta gets to share her love of technology with young people. She especially likes to help young black girls to join the technology field.

**Fun fact:** As a child, Peta loved to explore and tinker, like taking apart old clocks and then trying to put them back together.

## Activity 1: Network a neighborhood

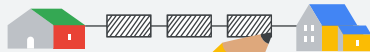
How does a map app determine the best way to get you from one place to another? Roads aren't always straight, and they can connect to each other in many different ways. For long trips, there can be thousands of ways to go. How do you know which one is best? Let's look at a real example of this kind of problem.

The phone lines in this town were blown down by a windstorm. The whole town is disconnected from the Internet. The mayor wants to install new high-speed Internet for everyone. She is asking you for help to design a single network that connects all the buildings to each other. To help the network cost less, it should use as few segments of cable as possible.



### Instructions:

1. You can start anywhere. Think about which paths between buildings will keep your cost low.
2. Connect buildings by filling in segments with a pencil or small counters - dry beans work great!
3. Check to make sure you can trace the path from any building to every other building.
4. Count the number of segments you used. That's your total length.
5. Can you find a different way to connect the buildings using fewer segments?



Length: 3

### Try it with someone else

There are a lot of different ways to approach this problem. See how someone else might solve it. It can be a classmate or someone who lives with you. Ask them to explain what they are thinking as they go. What do they do differently than you? What can you learn from each other?

Reflection



### Network a neighborhood

Think about how you solved this problem. What was difficult? How would you explain how you solved it to someone else?

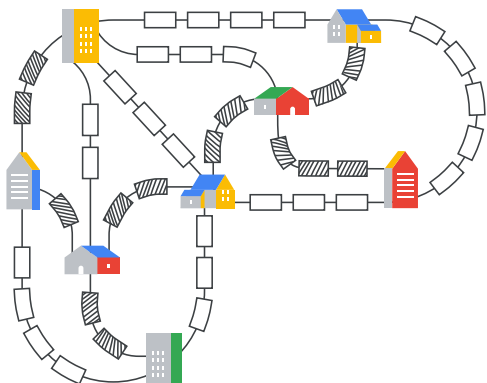
#### What was your best network?

Count up all of the segments. How many did you use to connect all of the buildings?

---

#### How does your solution compare?

There are a lot of different ways to solve this problem. Here's one. How is your map similar to or different from the one below?

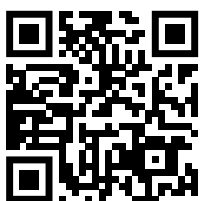


---

---

---

---



### Plug it in (optional) - Network a neighborhood

Try your path-making skills in this program in Scratch. You can generate new random maps and challenge yourself to get your path as small as possible.

[goo.gl/networkaneighborhood](https://goo.gl/networkaneighborhood)

## CS Career Connections



**Emily Henderson**  
(she / her)

### Mapping Maven

**Hometown:** Portland, OR

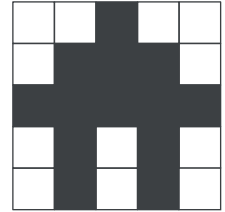
**Why her work is important:** Location is an important part of many of the world's challenges. People need good maps and data to respond to critical issues like earthquakes, hurricanes, and downed power lines. Emily has worked on issues ranging from traffic jams to COVID-19.

**What she does at Google:** Emily Henderson makes it easy for students and teachers to use Google's mapping technologies to learn about the world, write stories about what is important to them, and use data to build a better future.

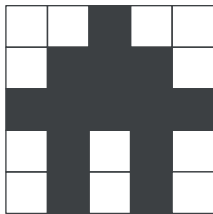
**Why she loves her job:** Emily loves to travel and explore the world, so working on Google Earth is a perfect fit!

## Activity 2: Encode an emoji

Every digital screen is made up of tiny squares called pixels. These pixels can be any color. Together, they make the images you see on your device! Look at the image to the right. It's a drawing of a house zoomed way in so that you can see the pixels.



A computer can't "see" images. It needs to know the color and position of each pixel to show a picture. One way to describe an image to a computer is to list the color of each pixel in order, like this:

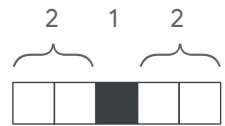


white, white, black, white, white  
 white, black, black, black, white  
 black, black, black, black, black  
 white, black, white, black, white  
 white, black, white, black, white

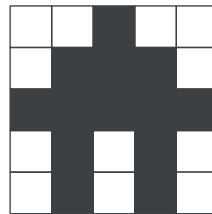


*That's a lot of words just to describe a tiny picture. A computer screen has millions of pixels.*

To make this easier, it helps to notice patterns in the pixels. For example, you could group pixels that are the same color together. The first row would read: 2 white, 1 black, 2 white. To make it even shorter, say 2, 1, 2.



This is a way of **encoding** a black and white image as numbers. Each number represents a group of pixels. Listing numbers takes up less space and time than listing every pixel color. Using groups saves space when large areas of an image are the same color. To turn that encoding into an image, read each row. Starting with white, fill in the pixels. Every time you see a new number, switch color.



2, 1, 2  
 1, 3, 1  
 0, 5  
 1, 1, 1, 1, 1  
 1, 1, 1, 1, 1

Instructions:

1. Start at the top-left corner. The first number in the encoding is the number of white pixels to start (even if it's 0).
2. Go to the next number in the encoding and color in that many black pixels.
3. Keep going through the encoding, switching between black and white pixels.
4. When you get to the end of a row, move to the next row down. Repeat until your image is done!

a.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|               |   |
|---------------|---|
| 2, 1, 2, 1, 2 | ✓ |
| 1, 2, 2, 2, 1 | ✓ |
| 0, 1, 6, 1    | ✓ |
| 8             |   |
| 1, 2, 5       |   |
| 3, 2, 3       |   |
| 5, 2, 1       |   |
| 8             |   |

b.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|                        |   |
|------------------------|---|
| 2, 4, 2                | ✓ |
| 1, 1, 4, 1, 1          | ✓ |
| 0, 1, 1, 1, 2, 1, 1, 1 |   |
| 0, 1, 1, 1, 2, 1, 1, 1 |   |
| 0, 1, 6, 1             |   |
| 0, 1, 1, 4, 1, 1       |   |
| 1, 1, 4, 1, 1          |   |
| 2, 4, 2                |   |

c.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|                     |
|---------------------|
| 3, 5                |
| 4, 4                |
| 4, 4                |
| 3, 2, 1, 2          |
| 0, 1, 1, 2, 2, 2    |
| 0, 1, 1, 3, 3       |
| 0, 1, 1, 3, 1, 1, 1 |
| 8                   |

d.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|                  |
|------------------|
| 0, 8             |
| 0, 1, 2, 2, 2, 1 |
| 8                |
| 8                |
| 8                |
| 0, 1, 6, 1       |
| 0, 2, 4, 2       |
| 0, 3, 2, 3       |

e.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

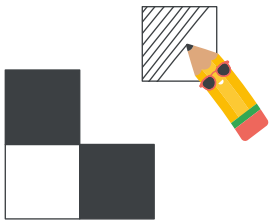
|               |
|---------------|
| 0, 2, 4, 2    |
| 0, 1, 6, 1    |
| 1, 2, 2, 2, 1 |
| 8             |
| 3, 2, 3       |
| 2, 1, 2, 1, 2 |
| 0, 1, 6, 1    |
| 0, 2, 4, 2    |

f.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|                     |
|---------------------|
| 8                   |
| 2, 1, 2, 1, 2       |
| 1, 1, 1, 2, 1, 1, 1 |
| 2, 1, 2, 1, 2       |
| 8                   |
| 2, 4, 2             |
| 1, 1, 4, 1, 1       |
| 2, 4, 2             |

**Create some of your own to share**



Reverse the process to create your own encodings to share with others!

1. Fill in pixels on the grid to create your image.
2. To encode, start at the top-left corner. Write down the numbers of white and black pixels, in order.
3. Repeat until you get to the end of the row. Then repeat for each row.
4. Send your encoding to a friend and see if they can draw your picture.

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|       |
|-------|
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|       |
|-------|
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|       |
|-------|
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|       |
|-------|
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |

Reflection



Encode an emoji

The images you just created may be quite small, but the same system also works for big images.

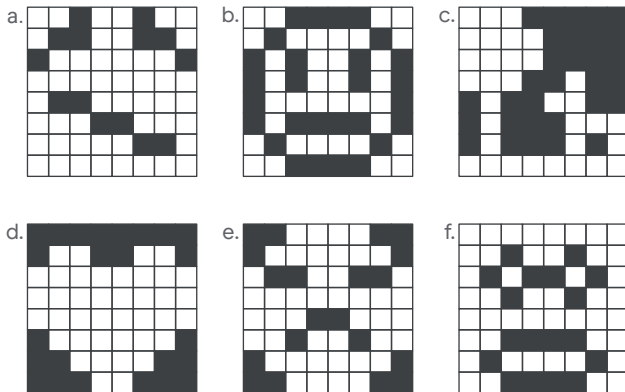
How could you encode more colors?

This encoding only works for black and white, but most emojis are full color. How could you improve this encoding to create color pictures too?

Five horizontal lines for writing a response.

Check your work

Compare your drawings with these solutions. How did you do?



CS Career Connections



Anthony D. Mays (he / him)

Data Dynamo

Hometown: Compton, CA

How he got his start: Anthony D. Mays first discovered computers and began teaching himself to code as a foster child at age eight. Starting early set him on a path to studying Computer Science in college, and eventually becoming a software engineer.

What he does at Google: Anthony builds software systems for data visualization. The tools he works on help people all over the world turn raw data into easier-to-understand charts and graphs. Just like emojis, data visualization can help communicate a range of complex ideas graphically.

Fun fact: Anthony uses a keyboard to code, of course, but his other keyboard is for playing music! He especially likes to record and share Animal Crossing cover medleys.



Plug it in (optional) - Encode an emoji

Next time you're on the computer, try using this Scratch app to recreate your own encoded images or to easily design new encoded images to share with others.

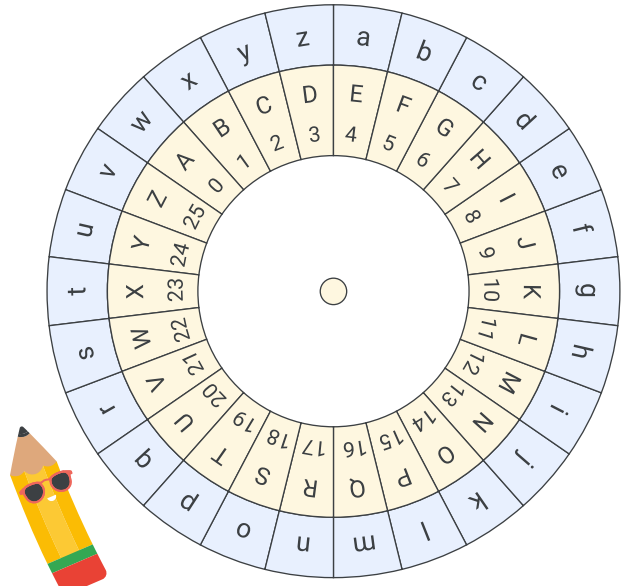
goo.gle/encodeanemoji



### Activity 3: Send a secret message

Some information on computers needs to stay secure. You definitely don't want everyone to know your password! How do you send information that can only be read by certain people? Long before computers were invented, people came up with tricks to make sure their private information stayed private.

**Encryption** is a way to hide a secret in plain sight. One very old encryption is called a shift cipher. Using a grid, you shift (or move) every letter in the alphabet a few spaces over. The number of spaces the letters were shifted is called the "key." If you know the key, then you can break the code to read the message. If you don't know the key, you can't understand the message. Look at this alphabet that has been shifted over four spaces. The key for this cipher is **4**.



|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |

Use the above chart to decrypt this message. Find each letter in the chart and write down the matching letter.

|   |   |   |  |   |   |   |   |   |  |   |   |  |   |   |   |   |   |   |  |  |  |  |  |  |  |
|---|---|---|--|---|---|---|---|---|--|---|---|--|---|---|---|---|---|---|--|--|--|--|--|--|--|
| y | o | u |  |   |   |   |   |   |  |   |   |  |   |   |   |   |   |   |  |  |  |  |  |  |  |
| C | S | Y |  | J | S | Y | R | H |  | Q | C |  | W | I | G | V | I | X |  |  |  |  |  |  |  |

There's a special tool for this kind of encryption called a **cipher wheel**. You can set the key for your cipher by spinning the wheels. When they are lined up right, you can encrypt and decrypt messages.

Instructions:

1. Remove the cipher wheel from the pocket on the back page of this booklet.
2. Turn the inner circle to line up the key number with the A on the outside circle.
3. Use your cipher to decrypt the following messages.
4. Turn the inner circle to line up the key number with the A on the outside circle.
5. Use your cipher to decrypt the following messages.

| Key | Encrypted Message     | Decrypted Message |
|-----|-----------------------|-------------------|
| 6   | lujotm oy iuur        | coding is cool    |
| 13  | v ybir gb uryc crbcyr |                   |
| 23  | ju ald fp qeb ybpq    |                   |
| 10  | s dkvu dy bylydc      |                   |
| 3   | orn zkdw l ghfrghg    |                   |

**Reflection**



**Send a secret message**

Use your cipher wheel to encrypt a message and send it to a friend to see if they can decrypt it!

**How could you keep your key secret?**

Your secret message isn't secret if everyone knows your key. How could you make sure other people can't figure out your key?

---



---



---



---



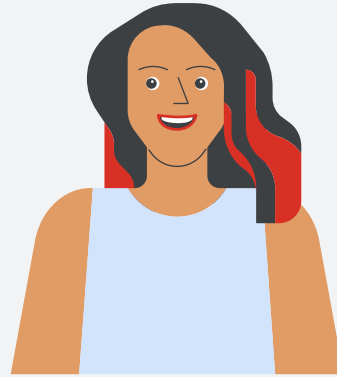
---

**Check your work**

Did you decrypt all of the messages?

| Key | Encrypted Message     | Decrypted Message     |
|-----|-----------------------|-----------------------|
| 6   | lujotm oy iuur        | coding is cool        |
| 13  | v ybir gb uryc crbcyr | i love to help people |
| 23  | jv ald fp qeb ybpq    | my dog is the best    |
| 10  | s dkvu dy bylydc      | i talk to robots      |
| 3   | ornn zkdw l ghfrghg   | look what i decoded   |

**CS Career Connections**



**Parisa Tabriz**  
(she / her)

**Security Princess**

**Hometown:** Chicago, IL

**How she got her start:** Parisa Tabriz first started coding in college. After her own website was hacked, she became fascinated with keeping web sites and computer systems secure.

**What she does at Google:** Parisa finds and fixes security problems to keep people's data safe, so she is affectionately known as Google's "Security Princess". She works on Google Chrome, a web browser that you've probably used! She has also worked with people who write TV shows and movies to help them make more accurate and diverse scripts.

**Fun fact:** When she's not hacking, Parisa likes to make things like art, food, or various do-it-yourself projects.



**PLUG IT IN (OPTIONAL) - Send a secret message**

This Scratch cipher wheel makes encrypting and decrypting your secret messages really easy! Just make sure the person you're sending messages with is using the same key.

[goo.gle/secretmessage](https://goo.gle/secretmessage)

*Network a Neighborhood* and *Encode an Emoji* activities were adapted from CS Unplugged materials, which is made available at [csunplugged.org](https://www.sunplugged.org) under a Creative Commons Attribution-ShareAlike 4.0 International license.

