Ciphering a sentence

DT + English

Years 3-4 Years 5-6

A **cipher** is a message that has been written in such a way (encoded) that it is unreadable by others. In this lesson, students will use mapping to encode a sentence. Students will work with a partner to create an **algorithm** that describes the encryption process. They will also examine encoded and decoded messages to **recognize** **patterns** to help decode messages and develop strategies to **decompose** the problem.

## Learning hook: Forming a strategy (10 minutes)

In this activity, students will think about strategies for sending messages that are guaranteed to be private.

|  |
| --- |
| **Pretend that you want to send a private message to a friend. The only way you can deliver the message is to write it on a board that is observable by anyone who passes by. What could you and your friend do to ensure that you are the only people who can understand the message?**  Think-Pair Share: Instruct students to think for one minute about how they might accomplish the task. Then pair students to discuss their ideas and to agree by consensus on a strategy that would work. After 2 or 3 minutes, invite students to share their ideas in a large group discussion. |

**Learning demonstration**: Create the alphabet mapping (25 minutes)

In this activity, students write rules for mapping and reordering the alphabet for encoding messages to create an [**algorithmic**](#_dnnkrn7kdbo5) cipher.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity:**  Use the rules below to create ciphers.  **Create two rules for mapping the alphabet.**   |  | | --- | | Sample rules:   * The characters are divided into two groups: (1) characters for which the image of their uppercase form has an enclosed area (such as P or O) and (2) characters that do not have an enclosed area in in the image of their uppercase form (such as I or Z). * Sort the two groups alphabetically, with group 1 first and then group 2. |   **Apply the rules.**   |  | | --- | | Applying rule 1:  Group 1: {A, B, D, O, P, Q, R} Group 2: {C, E, F, G, H, I, J, K, L, M, N, S, T, U, V, W, X, Y, Z}  Applying rule 2:  A, B, D, O, P, Q, R, C, E, F, G, H, I, J, K, L, M, N, S, T, U, V, W, X, Y, Z |   **Place the letter sequence that resulted from Rule 2 in the second row of the following table beneath the number 1 to 26 (example completed below).**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   **This table shows a one-to-one alphabet mapping (matching one letter with one number), Alphabet Mapping #1.**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | | A | B | D | O | P | Q | R | C | E | F | G | H | I | J | K | L | M | N | S | T | U | V | W | X | Y | Z |   **Now, invent a third rule and apply it to the above mapping to come up with another one-to-one alphabet mapping table. A third rule could be to place the even-number-mapped letters in alphabetical order followed by the odd-number-mapped letters in alphabetical order. Let’s call that Alphabet Mapping #2.**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1  1 | 1  2 | 1  3 | 1  4 | 1  5 | 1  6 | 1  7 | 1  8 | 1  9 | 2  0 | 2  1 | 2  2 | 2  3 | 2  4 | 2  5 | 2  6 | | B | O | Q | C | F | H | J | L | N | T | V | X | Z | A | D | P | R | E | G | I | K | M | S | U | W | Y | |

|  |
| --- |
| **Teaching Tips:**   * With older or more advanced groups, present the above steps to the class as an example in a large group before breaking students into pairs to create their own alphabet mappings. |

## Learning construction: Encode sentences (20 minutes)

In this activity, students will use the [**algorithm**](#_dnnkrn7kdbo5) mappings from Learning demonstration to encode sentences.

|  |  |
| --- | --- |
| **Activity:**  Divide students in groups of two and tell them the following:   1. **Think of a simple message you would like to send to your partner.** 2. **Encode it three times: first use just Alphabet Mapping #1, then use just Alphabet Mapping #2, and then use both mappings from the previous activity.**  |  | | --- | | For example, if the sentence is:  We drove to the gym.  Using the Alphabet Mapping #1, we find each letter of the sentence in the second row of the table created in Activity 1. We replace the letter with the corresponding number in the first row.  Please note that one must place letter breaks (the character “\_”) and separate each word by a blank space. In this example, the encoded sentence is:  23\_9 3\_7\_4\_22\_9 20\_4 20\_12\_9 11\_25\_17 | |

|  |
| --- |
| **Teaching Tips:**   * If students created their own mappings in the previous activity, have them use those mappings in this activity. |

## Reflection: Analysis and reflection (20 minutes)

In this activity, students will [**decompose**](#_dnnkrn7kdbo5) their own ciphers, how ciphers might be represented with a computer program, and some of the issues in breaking and designing ciphers ([**recognizing patterns**](#_dnnkrn7kdbo5)).

|  |
| --- |
| **Activity:**  Tell students to discuss and record answers to the following questions in their journals with their partners.   1. Which Alphabet Mapping (1 or 2) is easier to use to encode a message? 2. Are the rules used for Alphabet Mapping #1 clear and easy to understand? How could you improve the rules? 3. Is your new rule for Alphabet Mapping #2 clear and easy to understand? How could you improve your rule? 4. How would you implement your cipher in a computer program? Could you break down the problem and write the sequence of steps (an [**algorithm**](#_eb1b7wdmkcge)) that describes the process? 5. Imagine that another person gives you an encoded message and you do not know the mapping tables. How would you go about deciphering or decoding the sentences? What patterns would you look for to help you discover the one-to-one mapping? 6. If someone wants to come up with a cipher that is very difficult to for others to decode, what are some of the strategies that he or she should consider to break down the problem ([**decomposition**](#_dnnkrn7kdbo5))? |

|  |
| --- |
| **Assessment:**  Collect students’ reflections. Assess for thoughtful, complete responses. |

# Additional information and resources

## Lesson Vocabulary

|  |  |  |
| --- | --- | --- |
| **Term** | **Definition** | **For Additional Information** |
| **Encryption** | The process of encoding messages or information in such a way that only authorized parties can read it | <http://en.wikipedia.org/wiki/Encryption> |
| **Decryption** | The process of decoding messages or information in such a way that only authorized parties can read it | <http://en.wikipedia.org/wiki/Encryption> |
| **Cipher** | An encoded/encrypted message | <http://en.wikipedia.org/wiki/Cipher> |
| **Algorithm** | A step-by-step set of operations performed to solve a problem | [http://en.wikipedia.org/wiki/Algorithm](http://en.wikipedia.org/wiki/Encryption) |

## Computational Thinking concepts

|  |  |
| --- | --- |
| **Concept** | **Definition** |
| **Algorithm Design** | Creating an ordered series of instructions for solving similar problems |
| **Pattern Recognition** | Observing patterns and regularities in data |
| **Decomposition** | Breaking down data, processes or problems into smaller, manageable parts |

### **Curriculum links**

Links with Digital Technologies Curriculum Area

|  | | |
| --- | --- | --- |
| **Strand** | **Year** | **Content Description** |
| **Processes and Production Skills** | **3-4** | Define simple problems and describe and follow a sequence of steps (algorithms) needed to solve them [**(ACTDIP010)**](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACTDIP010) |
| 5-6 | Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) [**(ACTDIP019)**](https://www.australiancurriculum.edu.au/Search/?q=ACTDIP019) |

Links with English Curriculum Area

|  | | |
| --- | --- | --- |
| **Strand** | **Year** | **Content Description** |
| **Language: Phonics and word knowledge** | **Year 3**  **Year 4** | Understand how to apply knowledge of letter-sound relationships, syllables, and blending and segmenting to fluently read and write multisyllabic words with more complex letter patterns [(ACELA1826 - Scootle )](http://www.scootle.edu.au/ec/search?accContentId=ACELA1826)  Understand how to use phonic knowledge to read and write multisyllabic words with more complex letter combinations, including a variety of vowel sounds and known prefixes and suffixes [(ACELA1828 - Scootle )](http://www.scootle.edu.au/ec/search?accContentId=ACELA1828) |
| Year 5  Year 6 | Understand how to use phonic knowledge to read and write less familiar words that share common letter patterns but have different pronunciations  [(ACELA1829 - Scootle )](http://www.scootle.edu.au/ec/search?accContentId=ACELA1829)  Understand how to use phonic knowledge and accumulated understandings about blending, letter-sound relationships, common and uncommon letter patterns and phonic generalisations to read and write increasingly complex words [(ACELA1830 - Scootle )](http://www.scootle.edu.au/ec/search?accContentId=ACELA1830) |

Key words, Algorithm, Computational thinking, Secret messages, **Encryption, Decryption, Cipher, Pattern recognition, Decomposition**

## Administrative Details

|  |  |
| --- | --- |
| **Contact info** | For more info about Exploring Computational Thinking (ECT), visit the ECT website ([g.co/exploringCT](http://g.co/exploringCT)) |
| **Credits** | Developed by the Exploring Computational Thinking team at Google and reviewed by K-12 educators from around the world. |
| **Last updated on** | 22/01/2020 |
| **Copyright info** | Except as otherwise [noted](https://www.google.com/intl/en/policies/terms/), the content of this document is licensed under the [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0). |