Parity magic

▼ Learning outcomes

Students will be able to:

- Describe the steps it takes to work out how to find the card that is turned over.
  Computational Thinking: Decomposition
- Describe what are rows and what are columns.
  Mathematics: Numeracy
- Discuss why changing one card will change the state of the row and column which it is in.
  Mathematics: Numeracy
- Explain how knowing what odd and evens numbers are means you can solve the parity problem.
  Mathematics: Numeracy
- Explain that each card is a bit and that the cards can represent data.
  Computational Thinking: Algorithmic Thinking
- Explain why they chose each extra bit (card) when setting up a parity column/row.
  Computational Thinking: Generalising and Patterns

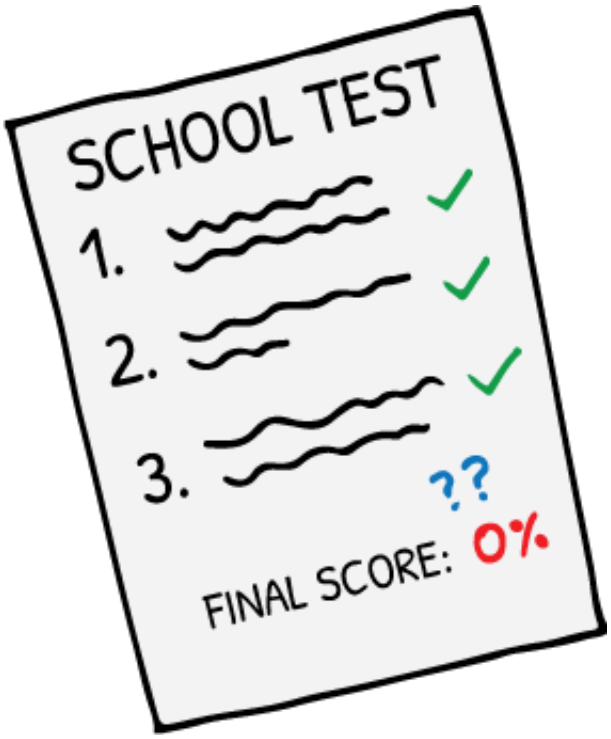
Key questions

- Why is it important for computers to be able to detect if the data received over the internet is the same as the data that was sent?
- What if I sent you an email that said you could now have Monday off

school, but when you received it, there was some electrical interference and a **bit** was changed from off to on so that the word "now" became "not". What would your reaction be?
- Can computers correct these sorts of mistakes automatically, and how would they do that?

Potential answers could include:

Students may come up with situations where computers won't read a disc or scan a code, but they may also confuse this kind of problem with other kinds of failures such as an error in a program or a flat battery.



Adults use computers for important things like banking, writing school reports, and communicating with each other. If the information being stored got changed without anyone knowing, you'd get the wrong balance in your account (too much or too little), the wrong grade in your report, or the wrong message in an email. Or worst still the website you are wanting to go to for your learning or the DVD you want to play won’t work! This activity will look at how computers correct this automatically.

Lesson starter

▼ See teaching this in action

CS Unplugged - Parity magic (sample classroom lesson)

vimeo.com/437726658?fl=pl&fe=cm

11:13

► Notes on resources

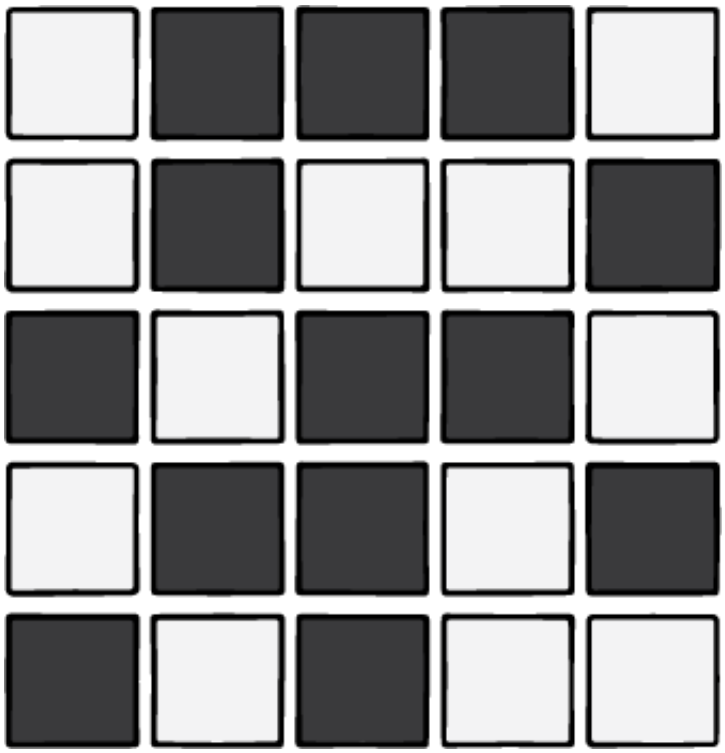
1. Teacher to class: "I've just learnt a magic trick I want to show you".
2. Teacher to class: "So who will be my assistant?"
3. Teacher to student: Hand the cards to the student and ask them to make a grid of 5 by 5 cards (calling it "5 rows of 5 cards might be clearer for younger students"). "Don't make any patterns - try to make it

as random as possible". To speed up the setup you could have two students do this task; encourage them to leave a small gap between the cards, and not be too fussy about making them straight.

4. Point out that the cards represent bits (binary digits). If they haven't studied binary numbers, you may need to just point out that this is the way everything is represented on computers - the cards here could represent a file, a message, a web page or even a program.

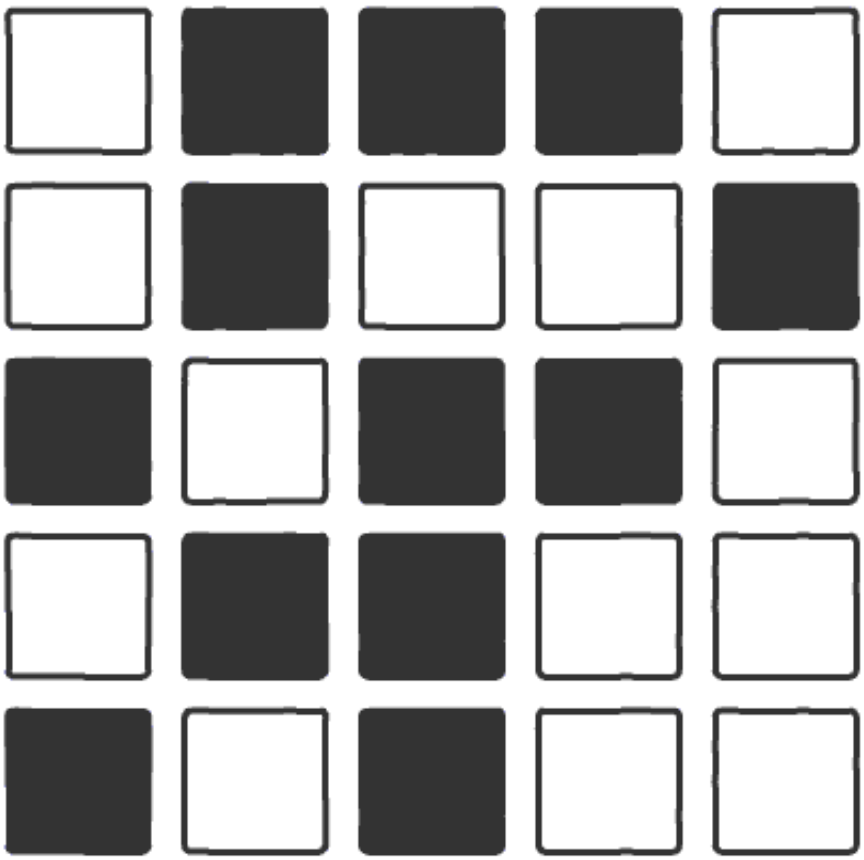
5. Teacher to class: "I'm going to make this a little bit harder by adding another row and another column".

► Teaching observations



Step 1: Example layout of a 5x5 grid set up by the volunteer.

Step by step adding a parity bit to each row and column



The last parity bit placed is useful because it will always work for both the column and row; if it doesn't match for both the row and column then you'll have made a mistake with one of the cards, and should go back and check them (try to not make it obvious that you're doing that).

▼ Teaching observations

Now you have added parity cards to make the number of black squares in every row and every column equal to an even number. Don't point this out to the students at this stage.

Lesson activities

Teacher to student: "I'd like you flip over one card while I cover my eyes".

Teacher to class: "Keep a close eye on which card it is to check if I have done my magic trick correctly!"

▼ Teaching observations

You need to emphasise that you want just one card flipped over. This prevents them from turning over too many cards (or none!). Usually students will follow this instruction, and their classmates will be able to confirm to you that, or let you know if one hasn't been turned over yet.

After the class confirms that a single card has been flipped, turn around to look at the cards. Scan the cards looking for the row with an odd number of black squares, and the column that has an odd number of black squares. The card that has been flipped will be at the intersection of these two lines. Turn this card over casually to restore it to the correct colour, saying “it’s this one”.

You can make a fuss that it might have been a fluke, so repeat the trick again. (After you put the card back to how it was originally, look away again and ask for another card to be flipped over.)

So is it magic? Or is it a trick?

Teacher to class: "Let’s first look at the cards before one was turned over" (make sure you've restored the card that was just flipped). The following steps will help the students to uncover what you've done:

- "Are there any patterns you can see? Think, pair, share".
- "Let’s break it down into parts".
- "Let’s look at the first row - count the black squares - how many are there?" 4 (in the example above).
- "Now the second row - count the black squares - how many are there?" 2.

- Note that it could be a row of all white squares - which would be 0 black squares; or it could be a row of all black squares - 6.

Teacher to class: “What do these numbers have in common?”. Students should be able to observe that they are all even numbers. If they need a hint, you could ask if there are any odd numbers there.

Let’s now look at the columns:

- "Does the first column have the same rule as the rows?".
- "What about the other columns?".

Teacher to class: “So how did we end up with an even number in every row and column - did the volunteer choose that?” (they didn't!).

Remove your extra row and column, and have them explain what colour to place at the end of the first row to make sure there is an even number of black cards. For example, if there are 3 black cards, ask what colour you need to add to make it into an even number (black)? If there are 4 black cards, they should work out that you need a white card to keep it even.

Continue doing this for each row; then do it for the columns. This is a good exercise for students thinking about even and odd numbers. For the last (corner) card, ask if you should use the row or column to decide it. They should observe that it's the same for both.

Once the extra row and column have been added, ask “So what happens when I turn a card over from black to white?” (It reduces the number of black cards by one, so it's now an odd number). "What if I change a white card to black?" (it adds one, which also gives an odd number).

The students might work out from this how to find the flipped card, but in either case, have a student come up, and ask them to look away while you flip a card. Then, when they look back, ask "Is the first row ok?" (They should notice that it's still even, so hasn't been changed). Carry on for each

row until they identify the one with an odd number of black cards. Draw a box around that row, and say "So one of these cards was flipped?". Now do the same with the columns - ask if each one is correct, then draw around the column that they identify.

Now ask "So which card was flipped?" Usually students will identify the one at the intersection.

▼ Teaching observations

Students have now worked out for themselves how this works. The key idea is that we just added a little more data, but could reconstruct the original if one card was changed.

▼ Mathematical links

This lesson would be useful for students learning about what even and odd numbers are.

It also exercises the idea of grids (you can use phrases like "5 by 5") and the language of columns and rows.

Applying what we have just learnt

- This kind of method is applied to almost all data that is stored or transmitted on computers (although usually a more sophisticated method is used that is even more reliable). If we didn’t have error detection and correction then unexpected errors in data would be common, and digital devices wouldn't be used to store anything important. The world would be in chaos and people wouldn’t trust computers. Computers wouldn’t be reliable.
- DVDs and CDs wouldn’t work if one fleck of dust was on the disc.
- Backing up wouldn't help much as this would also be unreliable too.
- Transmitting data over long distances (e.g. from space probes) would be particularly unreliable, since it can take minutes, or even days, for

data to arrive, and it's not feasible to request it to be retransmitted if it has had interference.

Lesson reflection

Try using other objects. Anything that has two easily distinguished 'states' is suitable. For example, you could use playing cards, coins (heads or tails) or cards with 0 or 1 printed on them (to relate to the binary number system).

What happens if two, or more, cards are flipped? (It is not possible to detect exactly which two cards were flipped, although it is possible to tell that something has been changed. You can usually narrow it down to one of two pairs of cards. With 4 flips it is possible that all the parity bits will be correct afterwards, and so the error could go undetected.)

Another interesting exercise is to consider the lower right-hand card. If you choose it to be the correct one for the column above, then will it be correct for the row to its left? (The answer is yes, always. In this card exercise we have used even parity—using an even number of black cards. Can we do it with odd parity? (This is possible, but the lower right-hand card only works out the same for its row and column if the numbers of rows and columns are both even or both odd. For example, a 5 × 9 layout will work fine, or a 4 × 6, but a 3 × 4 layout won't.)

Seeing the Computational Thinking connections

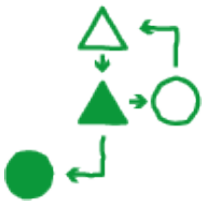
Throughout the lessons there are links to computational thinking. Below we've noted some general links that apply to this content.

Teaching computational thinking through CSUnplugged activities supports students to learn how to describe a problem, identify what are the important details they need to solve this problem, break it down into small logical steps so that they can then create a process which solves the problem, and then evaluate this process. These skills are transferable to any other curriculum

area, but are particularly relevant to developing digital systems and solving problems using the capabilities of computers.

These Computational Thinking concepts are all connected to each other and support each other, but it's important to note that not all aspects of Computational Thinking happen in every unit or lesson. We've highlighted the important connections for you to observe your students in action. For more background information on what our definition of Computational Thinking is see our notes about computational thinking.

▼ Algorithmic thinking



The step by step process of counting the number of black squares in each row and identifying whether the parity is even, and then repeating the process with each column, is an example of an algorithm. The full error detection algorithm includes what to to do when the error is found as well.

Examples of what you could look for:

Can students reliably find the flipped card in a grid? Can they articulate the algorithm to another student to help them complete the task?

▼ Abstraction

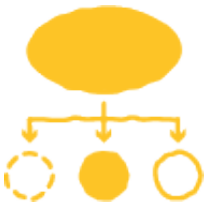
Each of the cards in this activity represents a bit inside a digital device. When we arrange all of our 'bits' together in the square this is a representation of some data (admittedly random data in the magic trick, but

it could be done with real data!). This means our grid is an abstraction because it is a model that could be used to represent any piece of data that can be stored on a digital device, for example a sound file, a video, or even a piece of code!

Examples of what you could look for:

Are students able to explain that each card is a bit and that the cards can represent data? Can they see the connection between the grid of cards and a piece of data made up of bytes?

▼ Decomposition



To accomplish this task students must decompose the process “find the error” into smaller steps. The first step students break it down to could be “look at each row, one by one, until I find an error”. Students can break it down further, focusing on one row, and asking themselves whether the row has an even or odd number of black cards. They are decomposing the bigger problem down to a subtask, which they can then work through and move closer to completing the full problem. Likewise when they need to find the column with an error they can break this down in the same way, focusing on one column, and asking the same question above: does the column have an even or odd number of black cards? These processes are much simpler to solve that the big problem “Find the error”, but if you work through them all then you will have completely solved that big problem!

Examples of what you could look for:

Are students able to break the task down into steps and describe these steps, without having to have the algorithm described to them first?

▼ Generalising and patterns

Once students understand the algorithm to find the card that has been turned, they will be able to find a (single) flipped card for any size grid; a 10 x 10 grid (100 cards) can be done fairly quickly, and even a 20 x 20 grid is possible given enough time (and cards!). They can generalise the problem "find the error in a **6 x 6 grid**" to: "find the error in **a grid**".

Examples of what you could look for:

Can students find the flipped card on larger grids?

▼ Evaluation

The parity trick is always able to detect and correct errors if one bit has been flipped, but it is important to evaluate this algorithm for different kinds of errors as well. If more than one card is flipped (i.e. more than one error occurs) then we can tell something is wrong with the data, but our algorithm can't tell us how to fix this! Even worse, if more than two bits are flipped then sometimes we might not even be able to detect the error! It's important that we evaluate this algorithm, because if it is going to fail sometimes we need to know. Computer Scientists evaluate algorithms like these and improve them when they find problems with them.

Examples of what you could look for:

Ask students "What kind of errors can be detected and corrected with the

parity cards? At what point can you only detect but not correct and error? Why is that?"

Can students explain what goes wrong when we try to detect the error if two cards are flipped?

▼ Logic

Flipping a card will always change a row/column from odd to even, no matter what the card is in what the other cards in the row/column are.

Also, the idea that the corner card is correct for both the new row and column is an advanced concept, but a pattern that some students might recognise.

Examples of what you could look for:

Can students explain why a single card flip always changes the number of black cards to an odd number?

Can students explain why the corner card will be correct for both the row and column? (This involves fairly advanced reasoning).