Error Detection and Parity Lesson Plan

Overview—*Error Detection and Parity*

Summary

When data is stored on a disk or transmitted from one computer to another, we usually assume that it doesn't get changed in the process. But sometimes things go wrong and the data is changed accidentally. This activity uses a magic trick to show how to detect when data has been corrupted, and to correct it.

Timeline

What	Time	Where
Demonstration	10	Parity.pdf
Exploring ASCII Worksheet	5-10	ParityWorksheets.pdf
Using Parity worksheet	5	ParityWorksheets.pdf
Error Passing Game	5	ParityWorksheets.pdf
Magic Trick Explanation	15	Parity.pdf
Wrap-up discussion	5	Parity.pdf

Materials

- ✓ 2 differently colored sets of 36 sticky notes
- ✓ Exploring ASCII worksheet, 1 per student
- ✓ Using Parity for Error Detection worksheet, 1 per student
- ✓ Error Passing Game worksheet, 1 per each group of 3 students
- ✓ 36 pennies for every 2 students (Optional)

Lesson Preparation

Before the lesson, you might want to watch the magic trick:

https://www.youtube.com/watch?v=voqghyZbZxo

and also the solution:

https://www.youtube.com/watch?v=gBPZOpT4DPU

Magic Trick- Whole Class

Here’s your chance to be a magician!

You will need a pile of identical, two-sided cards. (To make your own cut up a large sheet of card that is colored on one side only). For the demonstration it is easiest to use flat magnetic cards that have a different color on each side—fridge magnets are ideal.

Choose a child to lay out the cards in a 5 × 5 square (or set it up yourself), with a random mixture of sides showing, as shown in figure 1.

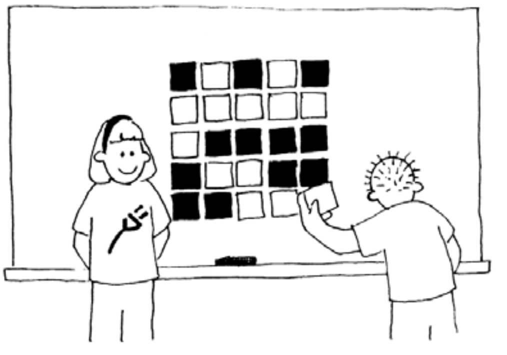

Figure 1: Student sets up 5x5 grid

Casually add another row and column, “just to make it a bit harder” (shown in figure 2).

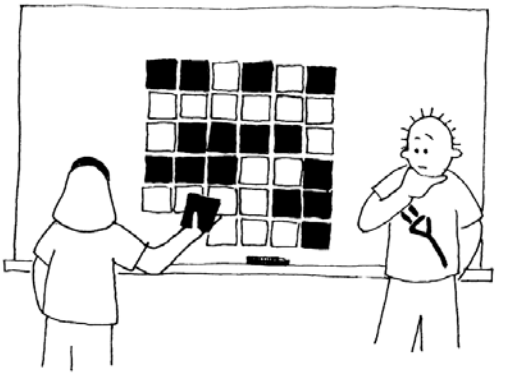

Figure 2: Parity row added to grid

These cards are the key to the trick. You must choose the extra cards to ensure that there is an even number of colored cards in each row and column.

Have another student volunteer flip over one card only while you cover your eyes (or leave the room if you can). The row and column containing the changed card will now have an odd number of colored cards, and this will identify the changed card.

Do not explain the magic trick! Tell the students it will be revealed soon.

Discussion– Whole Class

Ask the students whether computers are involved in sending information around the world. What sorts of files are sent? Have the students think about text-based information, like email, homework assignments, letters.

Ask the students whether computers are error-proof. Because computers are very fast at some tasks, and are good at things like arithmetic, it might seem like there are never errors. But when transferring information around, errors happen all the time. Computer scientists try to think up ways to detect errors (and sometimes even correct them).

Write the word TKGER on the board. Ask the students what word they think that should be (hopefully they all guess TIGER). Tell them that before we talk more about how computers detect errors, we want to talk about how computers represent letters and numbers.

ASCII. If you did the Binary Numbers or Image Representation activities, students should know that the computers store all data as numbers. Specifically, all of the text files, pictures, videos, and web pages are represented on your device using the binary numbers 0 and 1. These are referred to as bits (binary digits).

Many years ago computer scientists agreed to a standard way to represent letters and numbers. This encoding scheme is called ASCII (pronounced as ski), which stands for American Standard Code for Information Interchange. It's a system used to represent English characters. It uses 7 bits. Does anyone remember how many different binary numbers can be represented with 7 bits? [**Answer**: 128].

So ASCII includes 128 different characters, which includes numbers, lowercase letters, uppercase letters, and some special characters like punctuation marks.

Worksheet– ASCII and Parity

ASCII characters are stored using 8 bits (each ASCII character uses an 8-digit binary number). The leftmost seven bits in a binary number in ASCII are used to represent the character. For example, the letter “A” is number 65 in decimal, or 1000 001 in binary. The rightmost digit is used to represent a ***parity*** bit, which helps a computer know if an error has happened. The parity bit is used to make any number have an even count of 1’s.

Examples (show both, to avoid misconception that a 0 parity bit means correct, a 1 parity bit means an error):

- For our “A” example, the binary representation has two 1’s, so the eighth, rightmost bit will be a 0. The ASCII value for “A” would be 1000 001**0**.
- For a “C” the binary value is 67, which is 1000 011. This binary number has an *odd* number of 1 bits. So to ensure the 8-bit code has an *even* number of 1s, the parity bit must be a 1, i.e., 1000 011**1**.

Have students complete the Exploring ASCII worksheet.

Exploring ASCII Worksheet Key

Part I: Students should decode the word “BEGIN” with 7-bit numbers (no parity bit).

Part II: Students will create their own message using ASCII and swap with a partner. No specific answer.

Part III: The encoded message is “HDLLOBOB” but should be “HELLOBOB”. The code for the second character should be 1000 101.

Worksheet– Parity for Error Checking

Ask students how a computer might identify the error in the Exploring ASCII worksheet. Some students may have the idea of comparing to a dictionary. Although that's an interesting idea, the concept for this lesson is parity.

As shown in the worksheet chart, each character requires only seven binary bits (zeros and ones). To help identify errors, computer scientists came up with the idea of adding an eighth bit, known as the **parity bit**, to each character. Figure 3 shows the ASCII code for the letter A, with the parity bit added.

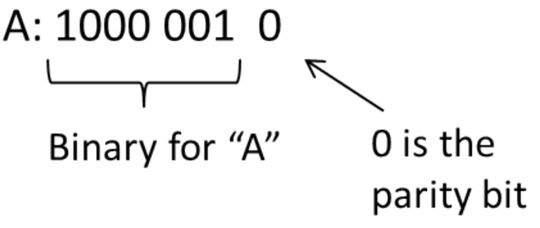

Figure 3: Letter A with parity bit

To practice, have students complete the worksheet Using Parity for Error Checking. Figure 3 is included at the top of that page as a reminder.

Using Parity for Error Checking Key

Part I:

a) How man bits (zeros and ones) are used to represent the letter C, without a parity bit? _**7**___
b) What parity bit is used for the letter C (circle one)? **1** or **0**
c) Why is 1 used as the parity bit for J, rather than 0?

The code for J is 1001 010, which includes three 1-bits (an odd number). So we must add a 1, so that the 8-bit code for J has an even number of 1-bits.

Part II:

W 1010 111	1	X 1011 000	1	Y 1011 001	0	Z 1011 010	0

Part III:

The highlighted character is wrong, because there are an odd number of 1 bits. Note that parity can detect that the character is wrong, but by itself it cannot correct the error. Error correction is a more advanced topic that requires other techniques. One possible option is to ask the sending computer to resend (with the idea that the same error would not occur twice).

```
1000 0100     1000 1011     1000 1110     1001 0010     1001 1100
```

Activity – Error Passing Game

This short activity reinforces the idea that the parity bit can help computers determine whether errors occurred. Students should work in teams of size 3 (if needed, a slightly larger group can be used, with more than one student as a message passer).

This activity is similar to the classic game of telephone.

- The first student (a computer) creates a 1-character “message” along with a correct parity bit.
- The message passer may either pass along the correct code or make exactly one error. The message passer is acting like the internet, sending characters from one computer to another.
- The third students is also acting as a computer, and trying to determine whether the character was passed along correctly.

Magic Trick Explanation – Whole Class

When students have completed the worksheet, discuss the limits of error correction. Note that some of these questions may have come up during the prior worksheets.

What happens if there is more than one error?
Answer: If two errors occur, an even number of “1’s” will be present, so we will *not* be able to detect the error!

After detecting the error, can the computer correct it?
Answer: No. The computer has no way of knowing which bit got flipped, so there is no way to get a valid character back.

How can we get a corrected character?
Answer: There are many different techniques. One students may come up with is to store the information in several places. This idea is called *redundancy* and is used in many real-world applications (Google stores our YouTube videos in several locations on Earth, so if one data center goes offline, we can still watch cat videos).

Teach the magic trick to the students:

1. At the board, go through the magic trick once more.
2. See if any students can guess how you did it.
    1. Start with the 5 × 5 grid
    2. Add a sixth card to each row and column, making sure the number of colored cards is always even (remember 0 is an even number)
    3. Have a student flip a card
    4. Look for a row with an odd number of colored cards. Circle it.
    5. Look for a column with an odd number of colored cards. Circle it.
    6. The flipped card should be at the intersection
3. Challenge the students to try the trick. They can use colored cards or even pennies, where some are heads and some are tails. Each pair of students will need 36 pennies.
4. Now take turns to perform the trick.
5. (optional) Encourage the students to try the trick at home with their families.

What's It All About Discussion – Whole Class

Imagine you are depositing $10 cash into your bank account. The teller types in the amount of the deposit, and it is sent to a central computer. But suppose some interference occurs on the line while the amount is being sent, and the code for $10 is changed to $1,000. No problem if you are the customer, but clearly a problem for the bank!

It is important to detect errors in transmitted data. So a receiving computer needs to check that the data coming to it has not been corrupted by some sort of electrical interference on the line. Sometimes the original data can be sent again when an error has been transmitted, but there are some situations when this is not feasible, for example if a disk or tape has been corrupted by exposure to magnetic or electrical radiation, by heat or by physical damage. If data is received from a deep space probe, it would be very tedious to wait for retransmission if an error had occurred! (It takes just over half an hour to get a radio signal from Jupiter when it is at its closest to Earth!)

We need to be able to recognize when the data has been corrupted *(error detection)* and to be able to reconstruct the original data *(error correction)*.

The same technique as was used in the "card flip" game is used on computers. By putting the bits into imaginary rows and columns, and adding parity bits to each row and column, we can not only detect if an error has occurred, but *where* it has occurred. The offending bit is changed back, and so we have performed error correction.

Of course computers often use more complex error control systems that are able to detect and correct multiple errors. The hard disk in a computer has a large amount of its space allocated to correcting errors so that it will work reliably even if parts of the disk fail. The systems used for this are closely related to the parity scheme. And computer scientists solve these puzzles, figuring out how to make our data reliable.