# GEMS Challenges ----- Binary Necklaces

| SUBJECT | Suggested grades | Cautions/concerns |
|---|---|---|
| Computer Science | 2-12 | Make sure the beads don't roll off the table |

**The Challenge—learn a little about binary numbers and why they are used in Computer Science**

| | LEADER ACTIONS | GIRL ACTIONS |
|---|---|---|
| INTRODUCTION | Present the challenge—you want to learn how to write your name in binary code like a computer does. | |
| BACKGROUND INFORMATION | Ask the girls what other words use the prefix bi. You should get answers like bicycle, biannual, bilingual, bisect, etc. Hopefully it doesn't get too far off track… :-)<br><br>Explain that computers understand all data as switches going on and off, and that the very simplest thing you will learn to code today is your name in binary. | Think about "bi" and words that use this prefix. |
| ACTIVITY | Today we are going to make necklaces with your names in binary code. This is a real code, and if you want to, you can use binary code to write secret messages to your friends. It is a little clunky though, and there are better codes for that. But the people here today are the only ones who will know what your necklace says.<br><br>Using the half sheet of paper, have the girls write the letters of their names going down the column marked Name. Use the code below it to color in the little circles that make the letters. Each 1 is colored in, each zero is not. This corresponds to a little switch being rapidly turned on and off.<br><br>When the names are completed, give each girl a piece of yarn/string, a cup of colored beads and a cup of white beads. | Fill out the name sheet.<br><br>Tie a knot in the end or your string. This is where the first letter of your name will start. Lay out white beads for 0, and |

| | LEADER ACTIONS | GIRL ACTIONS |
|---|---|---|
| | Have each girl choose the color(s) she wants to use for the 1s in her name, and a completely different color bead for the division between letters. Have them write this on the sheet or plate. Count the beads out on the small plates.<br><br>Ask the girls to tie a knot at the end of string to prevent spills and also to keep track of the direction she is working.<br><br>Begin to string the beads, working from the first letter of their names. Work from right to left.<br><br>When each girl is done, tie the necklace around her neck and be sure she can take it on and off easily. | choose a color to divide the letters, and a color for the letters. You will have three colors—white for 0, one color for dividing the letters, and another color for the 1s in your letters.<br><br>String your name. Work from right to left. Be sure to divide your letters with the third contrasting color bead.<br><br>When you are finished, tie your necklace around your neck, being sure it is loose enough to take off. |
| CONCLUSION | Imagine how data is converted so quickly now, using bits in 1s and 0s. | Read the handout about binary and bits |
| REFLECTION | Ask one or two girls to draw a card from the reflection deck and answer the question. | Reflect on what you learned, and how you might learn more. |

**Supplies per table**

Tub of multi-colored pony beads, tub of white pony beads

Yarn or cord—whatever is cheap—about a yard per girl

Worksheets for names

Small paper cups to hold the beads

Small paper plates to organize the beads

**Preparation needed**

- Print handout—Explaining BINARY
- Print name half sheet
- Do your name on the half-sheet as an example to show the girls before they start.
- Make a necklace ahead of time

**Comments**

Sorting the beads is tricky. Sometimes the numbers don’t work out so allow them to use white for 0 and a different color for 1 in each letter of their name and another one for the divider color.

NanoDays
The Biggest Event for the Smallest Science!

## My Name in Binary Code

**Computers use binary code to store information.** Binary code can be represented by any two symbols, such as 1s and 0s or full and empty circles.

Make a binary code representation of your name (or nickname). For each row, fill in a letter and the corresponding binary code. Leave the circle empty for 0 and fill it in for 1.

**Name** **Binary Code**

___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○

| Letter | Binary Code | Letter | Binary Code |
|---|---|---|---|
| a | 01100001 | n | 01101110 |
| b | 01100010 | o | 01101111 |
| c | 01100011 | p | 01110000 |
| d | 01100100 | q | 01110001 |
| e | 01100101 | r | 01110010 |
| f | 01100110 | s | 01110011 |
| g | 01100111 | t | 01110100 |
| h | 01101000 | u | 01110101 |
| i | 01101001 | v | 01110110 |
| j | 01101010 | w | 01110111 |
| k | 01101011 | x | 01111000 |
| l | 01101100 | y | 01111001 |
| m | 01101101 | z | 01111010 |

**Upper case letters start with 010** (instead of 011).



NanoDays
The Biggest Event for the Smallest Science!

## My Name in Binary Code

**Computers use binary code to store information.** Binary code can be represented by any two symbols, such as 1s and 0s or full and empty circles.

Make a binary code representation of your name (or nickname). For each row, fill in a letter and the corresponding binary code. Leave the circle empty for 0 and fill it in for 1.

**Name** **Binary Code**

___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○
___ ○ ○ ○ ○ ○ ○ ○ ○

| Letter | Binary Code | Letter | Binary Code |
|---|---|---|---|
| a | 01100001 | n | 01101110 |
| b | 01100010 | o | 01101111 |
| c | 01100011 | p | 01110000 |
| d | 01100100 | q | 01110001 |
| e | 01100101 | r | 01110010 |
| f | 01100110 | s | 01110011 |
| g | 01100111 | t | 01110100 |
| h | 01101000 | u | 01110101 |
| i | 01101001 | v | 01110110 |
| j | 01101010 | w | 01110111 |
| k | 01101011 | x | 01111000 |
| l | 01101100 | y | 01111001 |
| m | 01101101 | z | 01111010 |

**Upper case letters start with 010** (instead of 011).

whatisnano.org

Explaining Binary—a simple resource from https://mikkegoes.com/computer-science-binary-code-explained/

Back in the day when the very first computers were built, they had actual light bulbs to provide outputs to their users. So a user would see a certain light switched on to indicate a certain kind of output or message from the computer.

The same thing happens nowadays when you are driving your car and the gas light comes on. That is an output from your car, telling that you should get off the freeway and find a gas station ASAP.

So with a computer, these 1's and 0's can be pretty much anything in modern computers. But generally they represent numbers, letters, and other symbols.

The bottom line here is that **this simple concept of a switch being ON or OFF can translate into something really complex**.

Even the most sophisticated, modern computers all work according to this very basic, rudimentary machine language with the 1's and 0's representing two states: either ON or OFF.

But to make this happen, your computer obviously deals with a lot more than just a single switch being turned on or off.

Before we dive into how binary code and binary numbers actually work and how you can decode a simple binary sequence, let's consider one fundamental point about data storage first.

As mentioned above, computers take inputs to store and process information. This information or data is the fundamental ingredient for any computer to work.

Now, when you look inside a computer, you will see a bunch of circuits and electric wires. They carry all the information inside a computer, getting it to the right place to be either stored or processed.

**But how do you store or represent information using electricity?**

Now, the 1's and 0's we were just discussing represent the smallest unit of data that a computer understands.One switch being either on or off is what we need to store one **bit** of data.

Hence, a bit is the fundamental, basic unit of information. It is enough to indicate two different options, either "on" or "off". Also, it could mean "true" or "false", or simply "yes" or "no".

Having just two choices does not give us too many options really… But it is a good start!

With one light bulb we can store 1 bit of information. With 10 light bulbs, we could do 10 bits. So, if we had enough light bulbs, we could store any amount of data we wanted in a digital form.

Of course, computers use other methods and technologies than simple light bulbs to store data these days. Using light bulbs would not only take up too much space but it would also be difficult to store the data: turning off a computer using only light bulbs would mean that we would lose the data when the power goes off.

Thus, instead of light bulbs, computers store bits of data by holding electrons in capacitors, for example. Your computer uses this technology in its RAM memory.

**So, how many light bulbs would fit into your RAM exactly?**

Let's assume your computer has a 4GB RAM, for example. One GB is approximately a billion bytes. Or to be more exact, 1 GB is $2^{30}$ bytes. And 1 byte = 8 bits. That means your 4GB of RAM holds $2^{30}$ x 4 x 8 = 34,359,738,368 bits. **That's 34 billion light bulbs** – and we are only talking about your RAM here, not your 1TB hard drive!

So it's quite obvious that modern computers can do much more than decide whether to switch on a single light or not. Thus, even though binary code consists of only 1's and 0's, it can represent the most complex of computer programs these days.

With binary, we can use simple numbers to represent the different letters in the alphabet. So, "A" could be "1", "B" could be "2", and so on. That way, we can represent any word or paragraph of text as a sequence of these numbers. A computer can then store these numbers as information using the "on" or "off" signals.

So whenever you are reading text on your phone or your computer, what you see on your screen is based on binary code like this.

In a similar way to representing the alphabet in numbers, we can do the same thing for images and other graphic media, too. An image displayed on your screen consists of pixels. Each pixel in an image has a numerical value that determines the color it should display. Considering a single image can be made of millions of pixels, we're talking about a huge amount of information here!