

Challenge 247: It's a String Thing

A **binary string** is a list of numbers made up of 0's and 1's only.

0010 and 1011 are both examples of 4-digit binary strings.

There are exactly four different 2-digit binary strings: 00, 01, 10, 11

a Write down all eight 3-digit binary strings.

If you write the digits 0011 clockwise at the vertices of a square, and walk clockwise round the square calling out the digits as you come to them, you will recite all four 2-digit binary strings once each (though they overlap): first 00, then 01, then 11 and finally 10, as you return to your starting point.

Note that arrangements that are rotations of each other (that is, just with a different starting point) do not count as different, so in this 0110, 1100 and 1001 would count as the same arrangement as the one given.

A student was asked to show that there are only two ways of writing down 0's and 1's at the vertices of a regular octagon so that, as you walk round the octagon calling out the digits as you come to them, you will recite all eight 3-digit binary strings once each.

Here is the student's explanation of one of their attempts to do this, and why it fails.

I have got to 1110011? When you get to the question mark you have recited 111, 110, 100, 001, 011, but whether you write 0 (giving 110) or 1 (giving 111) next, you are repeating one of the 3-digit strings, and not including 000.

b Answer the same problem that was set to the student, by showing that there are only two ways of writing down 0's and 1's at the vertices of a regular octagon so that, as you walk round the octagon calling out the digits as you come to them, you will recite all eight 3-digit binary strings once each.

c Find a way of writing down 0's and 1's at the vertices of a regular hexadecagon (with sixteen vertices) so that, as you walk round the hexadecagon calling out the digits as you come to them, you will recite all sixteen 4-digit binary strings once each.

This idea has applications in security. For example, if you have to type a four-digit binary PIN number into a keypad of the type without an ENTER button that just reacts to the correct string of four digits and ignores incorrect strings, you would need to press $4 \times 16 = 64$ keys to try all the possibilities. However, if you enter the sequence you found in **c** you will try every possibility while only needing to enter 19 digits (you have to enter the first three again at the end to complete the circuit of all the 4-digit strings around the hexadecagon).

d If the keypad had 4-digit PIN numbers but allowed any digit from 0 to 9, how many presses would you have to make if you just tried every four-digit sequence one after another? How much time would you save if you could find a sequence of digits which went through every possible four-digit number once and only once?