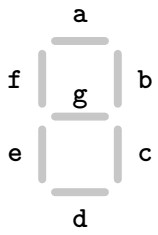
Problem 1.

A *seven-segment display* shows a digit by lighting up to seven little bars, labelled `a` to `g`. To draw a digit you switch some bars on and leave the rest off — so the whole display is really just **7 bits**: one bit per bar, where `1` means *on* and `0` means *off*. The seven bits for a digit are its row of the display's *truth table*.



Worked example — the digit 7. To make a **7** you light segments `a`, `b` and `c`; every other bar stays off.

`a b c d e f g = 1110000`

Read the bars in order `a`, `b`, …, `g` to get the seven output bits.

Your turn. For each digit below, shade the bars that light up, then write its seven output bits (`1` = lit, `0` = off).

digit	display	a	b	c	d	e	f	g
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								

Hint: it helps to picture the digit on the display first, then tick off each of the seven bars in turn.

Problem 2.

Advanced

A display chip can't read the shape "7" — it is handed the digit as a *binary number* and must switch on the right bars. (This is how a digital clock or a scoreboard lights up its digits.) Each digit arrives as binary on the *input*, and the seven segment bits come out as the *output*. So this time **both the input and the output are binary numbers.**

(a) Every digit must reach the display as a binary number. How many bits do you need to represent every digit from 0 to 9? _____

(b) Complete the truth table: write each digit as a 4-bit binary *input*, then the seven segment bits (`a`–`g`) it switches on as the *output*.

decimal	input (binary) 8	4	2	1	output (segments) a	b	c	d	e	f	g
0											
1											
2											
3											
4											
5											
6											
7											
8											
9											