CS Unplugged is a website where you can find many exercises that deal with programming and computational thinking. You can find it at http://csunplugged.org/.

The exercises do not require a computer. Many of them are quite similar to the mathematical tasks in this guide, but presented from a programmer's viewpoint. The following task is modified from the Binary Numbers task on the CS Unplugged website.

BINARY NUMBERS

Materials:

- **copies of the Binary Numbers template, one for each participant (the template can be found after these instructions)**
- **scissors**
- **a large empty sheet of paper and a marker pen for the instructor**

The idea of the task:

In this task we try out representing numbers in binary format, that is, in base-2 numeral system. While in the decimal system we can use the numbers 0-9, in the binary system we only have 0 and 1. This is also the idea behind the bits used by computers - a bit is either off or on, i.e. 0 or 1. If we want to write the decimal system number 2, one bit is not enough. In the binary system it is written as 10.

The cards cut out by the participants represent bits. Bits can be either on, with the side with dots facing up (1), or off, with the blank side up (0). When you count the number of dots facing up, you have the decimal value represented by the bits on the table.

Instructions for the task:

1. Participants cut the 8 rectangles on their sheet into separate cards. We will first use only 5 cards, so put aside cards F, G and H.

2. Each participant sets cards A-E in a row on the table in front of them so that they read **EDCBA** from left to right. The cards should be set so that there is more dots on the left and only a single dot on the rightmost card.

3. All cards are turned around so that the blank side is facing up (but their positions on the table do not change!).

4. As a warm-up, let's start counting up from 0. The instructor calls out the decimal numbers, then writes down the decimal number and the binary number formed by the cards side by side on the large sheet.

0 - do nothing to the cards, as there are only blank sides facing up. The binary number is 0, or using all available bits, 00000.

1 - turn the rightmost card (A) face up. One dot is visible. The binary number is 00001.

2 - the rightmost card is not enough for this! Turn over the next card to the left (B). Now there are three dots, so turn card A back face down. Two dots are visible. The binary number is 00010.

3 - turn the rightmost card back face up. Three dots are visible. The binary number is 00011.

4 - we need yet another bit! Turn the next card to the left face up (C). Now there are seven dots visible. Turn cards A and B back face down. Four dots are visible. The binary number is 00100.

Continue in this fashion until all participants seem confident in turning the cards around.

5. Ask the participants what some familiar decimal system numbers are in binary format. For example 7, 11, 16 or the participant's age.

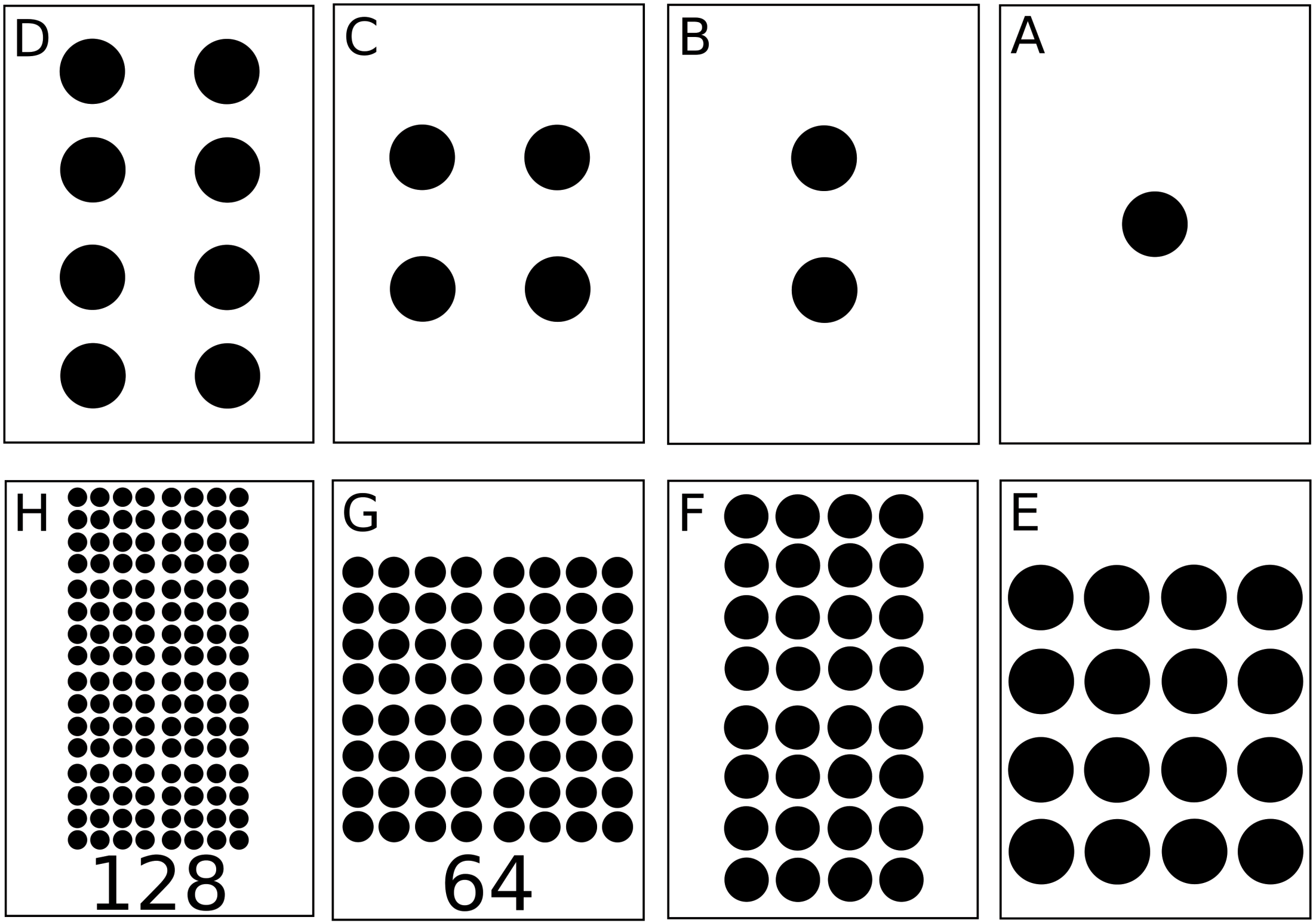
6. Ask the participants to write the decimal number 31 in binary format. (Answer: 11111)

7. Ask the participants to write the decimal number 32 in binary format. (Answer: we need yet another bit! Let's put the card F to the left of the other cards. It has 32 dots, so the other cards must be turned face down. The binary representation is 100000.)

8. If the participants feel up to the task, you can also include cards G and H. With 8 cards we have a byte of space at our disposal. A byte is a basic unit within the computer, used for example to denote hard drive size. One gigabyte is a billion bytes, that is, 8 billion bits.

Here is a table for checking some conversions from decimal to binary system.

Decimal	Binary	Decimal	Binary	Decimal	Binary
1	00000001	11	00001011	30	00011110
2	00000010	12	00001100	40	00101000
3	00000011	13	00001101	50	00110010
4	00000100	14	00001110	60	00111100
5	00000101	15	00001111	70	01000110
6	00000110	16	00010000	80	01010000
7	00000111	17	00010001	90	01011010
8	00001000	18	00010010	100	01100100
9	00001001	19	00010011	128	10000000
10	00001010	20	00010100	255	11111111


D
C
B
A
H
128
G
64
F
E