On the Subject of Needle Movements

To the left! Take it back now, yall!

(If the module doesn't make a series of glitched noises when you press the top left button or the needle doesn't spin on its own, you're looking at the wrong module.)
On the module is a spinning needle and three buttons. There will be one button on the top left of the module and two on the top right. To start, press the top left button to get your starting position in the grid below.

49	67	80	96	29	49
94	12	73	28	38	83
16	19	96	62	15	42
40	22	24	20	72	21
83	47	36	41	62	99
84	82	98	28	90	42

The needle will be spinning and indicating a series of directions to move. Use the directions it stops on, with one full spin indicating to go the same direction and two spins indicating the start of the sequence, to move around the grid. The grid wraps around. Multiply together each number you land on and convert it to binary, not counting the number you started at. If the binary string is less than 24 digits, repeat it until you reach 24 digits. If the binary string is more than 24 digits, use only the first 24 digits. Input this binary using the top right buttons, with the left button as 0 and the right button as 1, to solve the module. Pressing the wrong button will cause a strike and will require you to restart inputting the answer.

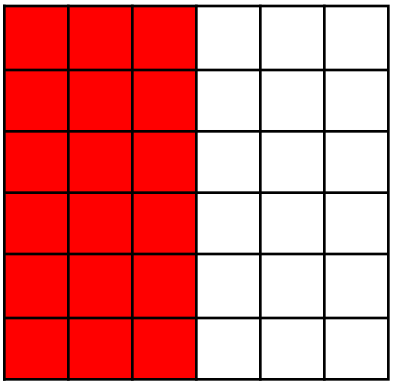
A note from the Department of Module Security and Assembly

I swear, someone's getting fired. First those half finished who's on first bootlegs and now this?

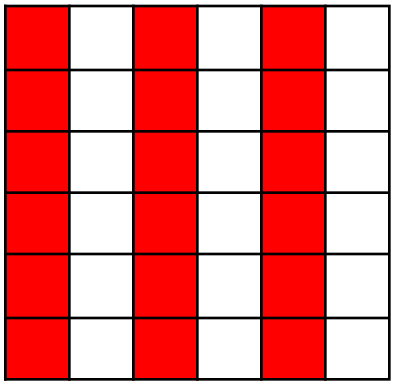
Alright, so I'm kinda rushing this here since the faulty bombs already got shipped and there's not much time before they go off so I don't have much time to polish this section here. The module got corrupted and what you just read isn't completely accurate. We were making these modules, they were supposed to be lookalikes, kinda like the not ___ series. But there was this fire, I think, nobody really wrote down why these got damaged, and it burned through the assembly line, which damaged the electronics and stuff. Nobody really noticed it for a while and by the time someone did they were already on bombs and sent out. The code of the modules were all messed up, but luckily they were in a predictable enough way that you can still solve them.

Alright, so the first problem is with the starting coordinates. It's supposed to say where you start at, but instead it outputs this garbled mess. Luckily you can still figure out where to start. The glitched sounds are specific to where the starting point in the grid is with a pause between each of them. Find each glitch noise played below. The red cells in the corresponding tables match to where the starting point could be.

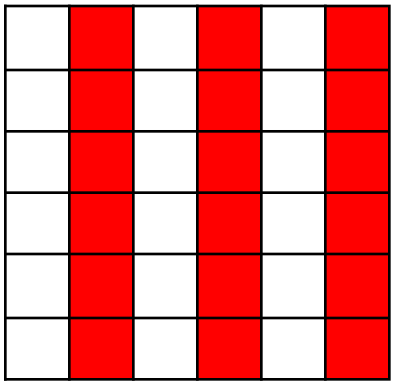
Sound here



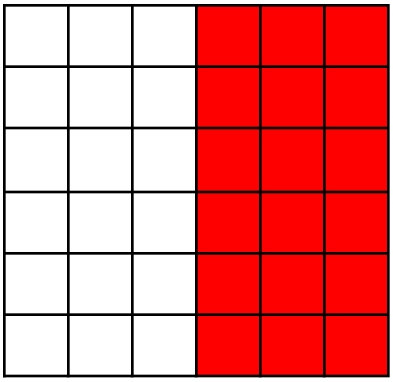
Sound here



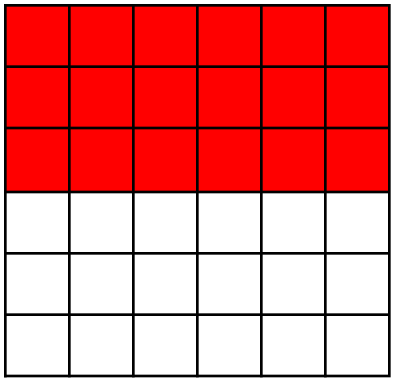
Sound here



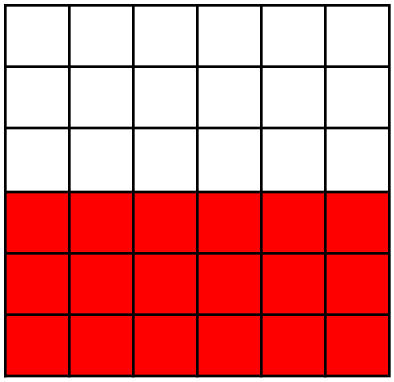
Sound here



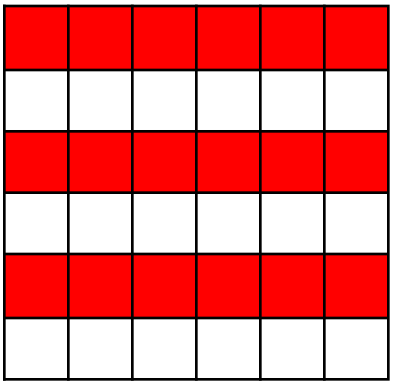
Sound here



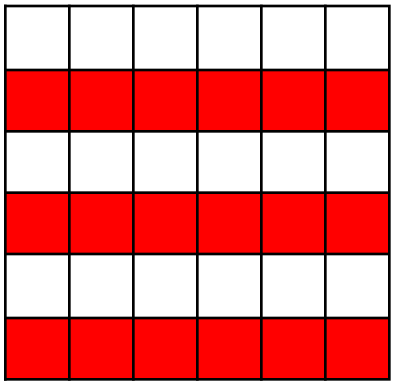
Sound here



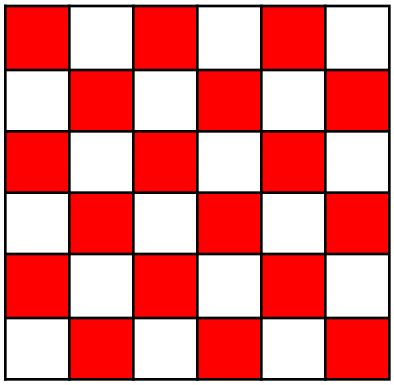
Sound here

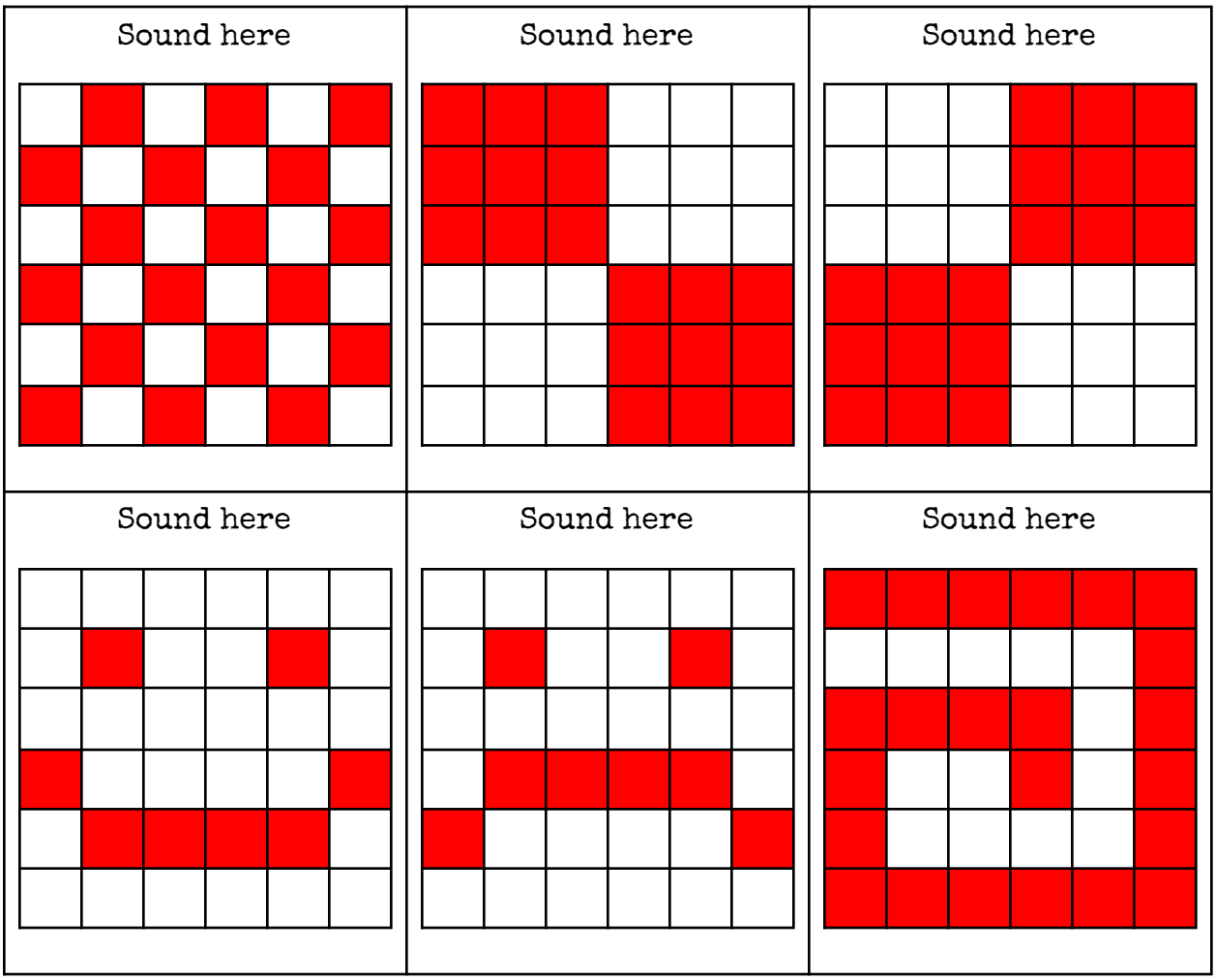


Sound here



Sound here





There's still another glitch you need to deal with. Normally these modules are supposed to be self contained, but someone wanted theirs to be a boss module. I'm not sure if it ever got finished, but what that meant was the boss chip was now in the module allowing it to access other module data, allowing it to call the functions of other modules. It should've been just their module, but we only had a week to finish all of these, so they ended up in the other modules too, and now this module is going to start using other code to modify the answer you got. Remember that 24 digit binary number you got? Split that up into groups of four. Each group will be modified according to its matching digit in the serial number where the first group is the first digit, second group is the second, and so on. Note each group doesn't have to be four digits long. If an instruction asks you about other groups, it means the group before any modifications have been done.

- 0: Do nothing.

- 1: Reverse the string.
- 2: Bitflip the string. (0 becomes 1, 1 becomes 0)
- 3: Swap the order of the first and second digits and the third and fourth digits.
- 4: The first digit is the result of XORing the first and second digits, and the second digit is the result of XORing the third and fourth digits. The last two digits are discarded.
- 5: Bitflip the first and last digits.
- 6: Repeat each digit of the string twice. (For example 1010 becomes 11001100)
- 7: Treat the string as if it was in decimal, then convert that to binary.
- 8: Take the last two digits and move them to the start.
- 9: XOR this with the next group, or the first group if this is the last group. This group becomes the results of that. (For example 1100 with the next group being 1010 makes it 0110)
- A: This group becomes the binary representation of the number of ones in the group.
- B: This group becomes the binary representation of the number of zeroes in the group.
- C: This group becomes the binary representation of the number in the cell you started at.
- D: This group becomes the binary representation of the row you started at, numbered top to bottom.
- E: This group becomes the binary representation of the column you started at, numbered left to right.
- F: Convert this group to decimal, add it with the cell you started at, take the digital root of that, then convert it back to binary.
- G: Convert this group to decimal, multiply it with the cell you started at, take the digital root of that, then convert it back to binary.
- H: Move all the ones to the start of the string.
- I: Move all the ones to the end of the string.
- J: Convert this string to decimal, add the numbers in the serial number, then convert this back to binary.
- K: Look at the next character in the serial number and convert it to binary as if it was a base 36 number.
- L: Remove any consecutive digits in the group. (For example, 0110 becomes 010)

- M: Convert the number to decimal, add it to the cell you ended at, average each digit together, remove the decimal point and any digits past the thousandth place, then convert it back into binary.
- N: The new group is the second digit, then the third digit, then the first digit, then the fourth digit.
- P: Remove all the zeroes.
- Q: Remove all the ones.
- R: Reverse the string then bitflip it.
- S: Convert the number to decimal, order the digits in ascending order, then convert it back to binary.
- T: Convert the number to decimal, order the digits in descending order, then convert it back to binary.
- U: Create a two digit number where the first number is the number of zeroes and the second number is the number of ones. Convert this back to binary.
- V: Create a two digit number where the first number is the number of ones and the second number is the number of zeroes. Convert this back to binary.
- W: Take each direction the needle pointed at, where 1 is the topmost and continuing clockwise. Convert this number to binary and add it to the decimal representation of this string. This is the new string for this part.
- X: Add the number of glitched sounds to the decimal version of this string. Convert the result back to binary.
- Z: Take the position of the glitched sounds in the table above, with the top left corresponding to the first and going in reading order. Add them all together. This is the new string for this part.

After you've modified each group, concatenate them all back together and input it into the module.