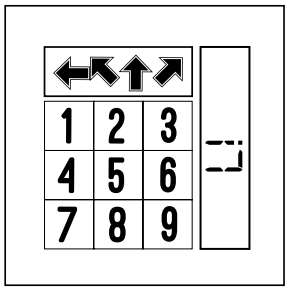
On the Subject of Keypad Directionality



Am I actually going anywhere with these arrows?

You are presented with a unique keypad. It uses arrows for the generation of the code. It has a feature to generate multiple layers of code based on the number of modules available on the bomb that it can process.

The starting number to generate every code is normally hidden in the keypad. However, there is a bug on the current keypad model that causes it to miscolor the starting number for the first layer of the keypad. The starting number will be indicated by it being a different color than every number on the keypad.

To generate the code for the first layer, use the starting number and move your position in the 3 by 3 table below. The direction you must take in the table is based on the direction of the arrows shown in the display in reading order. If you will move out of the table while moving, loop over to the other side of the table. Keep track of the numbers that you encounter on every movement in the table.

After the 4 movements, combine every number you encountered in proper reading order into a 4 digit number. This number is the code for the first layer.

Every time an eligible module in the bomb is solved, you will move to the next layer with the arrows having new orientations.

To generate the code after the first layer, use the last digit of the previous code as your starting number and repeat the steps performed on the first layer. Repeat the previous step with further layers until all eligible modules have been solved.

After all the eligible modules have been solved, the arrows will disappear and the display beside the keypad will show **Layer 1.** The keypad will now accept input. Type the code that you gathered on the layer shown on the side display.

If the 4 digit numbers you typed is correct, you will move forward to the next layer. If not, an alarm will be sounded, and a strike will be issued.

Repeat the previous step until you solve every layer on the keypad. After that, the keypad will deactivate, and the module will be solved.

The Table

1	2	3
4	5	6
7	8	9

Note:

If the module amount it can process is 0, the keypad will throw an error since **Layer 0** is not a part of its layer count. The module will activate a bypass and automatically unlock itself.