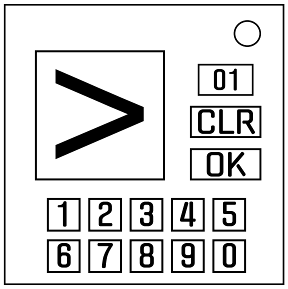
On the Subject of Brainf***



```
-[--->+<]>---.[------>+<]>--.--[--->+<]>-.+++[->+++<]>.-.-[--->+<]>-.[-->+++<]>+.+.-
-[--->++<]>.++[->+++<]>.+++++++++.+++.[-->+++++<]>+++.+[----->+<]>.----.+++.-------
-.-[--->+<]>-.[->+++<]>++.-.+++.-[--->+<]>-.++[--->++<]>.-----------.+++++++++++++.-
------.--[--->+<]>--.[->+++<]>++.++++++.--.[->+++<]>-.
```

- This module is a modified vesion of the famous esoteric coding language called Brainf***, which only consists of 8 characters. Refer to the table below for what each character does.
- Brainf*** works by **moving a pointer on a 1D array** and using commands to change the value of the selected cell. The pointer starts at **position** 0 and can't move onto negative positions or make numbers negative. All values are **set to** 0 at the start.
- The module consists of a display, a 10 button keypad, a delete button and a submit button. The last two digits of the stage number are shown on the smaller screen on the right.
- Every time a module is solved*, a new character in the program will be displayed on the screen, as well as the stage number being updated. Take note of the character and add it to the end of the program.
- After all modules are solved*, a fullstop will appear on the screen and the stage number will disappear. When this happens you must input the value of the current cell you are on **modulo 100** and press OK to solve the module. Inputting the incorrect number will give you a strike.
- A fullstop may also appear before then. If that happens you must input the value of the current cell you are on **modulo 100** and press OK before solving anymore modules. Inputting the wrong number or solving another module before submitting will give you a strike.
- Numbers should not be padded out with 0s. You will receive a strike if you do this.

Some modules are ignored by Brainf*.*

The table below

Symbol	Description
>	Moves the pointer right by one.
<	Moves the pointer left by one (unless the pointer is at position zero).
+	Increments the selected value by one.
-	Decrements the selected value by one (unless it's zero).
[and]	Anything inside the brackets is looped until the selected value at the end of the loop equals zero.
.	Outputs the current value of the selected cell.

,	Takes the input of the user and stores it the selected cell.

The Modifications

Looping

The origanal rules of looping still apply. However, to prevent the values from exceeding the assigned memory slot, each loop can repeat no more than x times where x = batteries + 1

The comma

Due to the National Regualtions on Bomb Creations, Part 28, subsection 6, the defuser will not be able to assign their own values to cells when a comma appears. Instead the value of y must be assigned to the selected cell. To get y:

1. Take the value of the cell to the left of you (if you are in position 0 take the one to the right of you).
2. Modulo the value by 6 and add 1 to it.
3. Take the character in the serial number that is in the position indicated by the number you just got.
4. If it is a letter turn it into it's position in the alphabet (A=1, B=2, e.t.c).
5. That is the value of y.

Going negative

- If you are about to go to a negative position, go right instead of left.
- If you are about to create a negative number, add one instead of subtracting.