Operation	Name	Usage
$N\ k\ n$	Number	**Initializes the memory variable** v_k to **have the value** n. Number cards are used to specify constants used in the calculation and usually appear at the beginning of the program.
+	Add	**Sets the Analytical Engine into addition mode.** When the operands are loaded into the ingress column, the two values are added together and placed on the egress column. If the result cannot fit in the number of digits available, the runup lever is set.
-	Subtract	**Sets the Analytical Engine into subtraction mode.** When the operands are loaded into the ingress column, the value in I_2 is subtracted from I_1 and the result is placed on the egress column. If the result is negative, the runup lever is set.
*	Multiply	**Sets the Analytical Engine into multiplication mode.** When the operands are loaded into the ingress column, the two values are multiplied together. Because the result of a multiplication is likely to be much larger than the input values, the output is stored in two columns that together act as a column with twice the usual number of digits. The first half of the result is stored in egress column E' and the second half is stored in egress column E. As long as the numbers are small, you can ignore E' and take the result from E, just as you would for addition and subtraction. The runup lever is not affected.
/	Divide	**Sets the Analytical Engine into division mode.** The number contained in the combined columns I_1' and I_1 is divided by the contents of I_2. The quotient is placed in E' and the remainder is placed in E. If the quotient does not fit in its column or the divisor is zero, the runup lever is set.
$L\ k$	Load	**Copies the contents of variable v_k into the appropriate ingress column** , which is I_1 for the first such instruction and I_2 for the second L instruction in a pair. The contents of variable v_k are not affected. If the L in the instruction is followed by a single quotation mark, the value is copied into I_1'.
$S\ k$	Store	**Stores the value from the egress column into variable v_k.** If the S is followed by a single quotation mark, the value is taken from E' instead.
$P\ k$	Print	**Print Outputs the value from variable to the printer.**

Operation	Name	Usage
$B\,n$	Backward	**Moves the card reader backward n cards ,** allowing the machine to repeat previous operations. In counting the number of cards to move, it is important to keep in mind that the current operation has already been read.
$?F\,n$ $?B\,n$	Conditional	These operations are similar to the F and B operations except that the card reader moves only if the runup lever is set.
$F\,n$	Forward	**Moves the card reader forward n cards, skipping those operations.** Note that the current card has already been read, so that the instruction $F\,1$ skips the next card.

Attentions

> Some Hints for Code/Program && Simulator

- The simulator will be interrupted if the program executes **more than 10^4 steps**.
- The **initial value** of each column is 0.
- You can add **Single-line comments** to your code. The comment is started with `#` and continues until the end of the line. Please note that Multi-line comment is not supported.
- If a step doesn't set the runup lever, it will **reset** the runup lever to 0.
- Loading a variable to an ingress column is **NOT** allowed before the mode is set to "+", "-", "*" and "/".
- Loading a variable to I_1' is **NOT** allowed if the mode is not set to "/".
- Any comment and blank line will be **ignored** by the simulator, that is, a blank line in your code **will not be regarded as a real line** in the program. So be careful when using `B`, `F` , `?B` and `?F`. You can make the most of the **simulator** to help you understand how it works on Conditional instructions.

Usage of the simulator:

Button	Single Click	Long Press
`<`	Backward to the last step	Go back some steps
`▷`	Start Simulation	Adjust the speed of simulation
`□`	Stop and return to the initial state	View the current step and the total steps
`>`	Goto to the next step	Go forward some steps

> v for variable, I for ingress, E for egress