The Instructo Paper Computer
Instruction Set Reference
(adapted from Operator's Manual page 12)

In each instruction:

`*` is either `A` or `B` (or `C` for the jump switches)

`SS` is a memory address: either some main storage address `90`–`99`, or a program line `00`–`89`, or `IN` for the index counter.

`#` is either 0 or 1, referring to a jump switch value

Input/Output

`ENI*, SS`	Enter value from Input `*` into memory location `SS`
`PRO*, SS`	Print from memory location `SS` to Output `*`
`PAB*, SS`	Print a mixed fraction on Output `*`. The whole number part comes from Register A, the numerator from Register B, and the denominator from memory location `SS`
`PBA*, SS`	Print a mixed fraction on Output `*`. The whole number part comes from Register B, the numerator from Register A, and the denominator from memory location `SS`

Registers

`LDR*, SS`	Load the number in memory location `SS` into Register `*`
`STR*, SS`	Store the number in Register `*` into memory location `SS`
`SWAP`	Swap the values in Registers A and B

Math functions

`ADD*, SS`	Add the number in memory location `SS` into Register `*`
`SUB*, SS`	Subtract the number in memory location `SS` from Register `*`
`MUL*, SS`	Multiply the number in memory location `SS` into Register `*`
`DVD*, SS`	Divide Register `*` by the number in memory location `SS`. Answer is a decimal
`DIV*, SS`	Divide Register `*` by the number in memory location `SS`. Stores quotient in this register, and remainder in the other.
`EXP*, SS`	Raise Register `*` to the exponent from memory location `SS`
`SQT*`	Take the square root of Register `*`
`DRT*`	Take the digital root of Register `*`
`REV*`	Reverse the digits of Register `*`

Index counter

`INDL, SS`	Load the number in memory location `SS` into the Index Counter
`INDA, SS`	Add the number in memory location `SS` into the Index Counter
`INDS, SS`	Subtract the number in memory location `SS` from the Index Counter

Jump Switches & Compare Unit

`SJ*#, SS`	Set Jump Switch `*` to value `#`
`CPR*, SS`	Set up comparison between Register `*` and memory location `SS`. The next line should be one of the compare unit Jump instructions

Program flow (execution jumping)

`JUMP, SS`	Jump to program line `SS`
`J*NZ, SS`	Jump to program line `SS` if memory location `SS`is nonzero
`J*ZE, SS`	Jump to program line `SS` if memory location `SS`is zero
`JIBD, SS`	Jump to program line `SS` if there is data showing in Input B
`JJ*#, SS`	Jump to program line `SS` if Jump Switch `#`has value #
`J*EQ, SS`	Jump to program line `SS` if values in the Compare Unit `*` are equal
`J*NE, SS`	Jump to program line `SS` if values in the Compare Unit `*` are not equal
`J*LT, SS`	Jump to program line `SS` if Register `*` is less than the other value in Compare Unit `*`
`J*GT, SS`	Jump to program line `SS` if Register `*` is greater than the other value in Compare Unit `*`
`J*NL, SS`	Jump to program line `SS` if Register `*` is not less than the other value in Compare Unit `*`
`J*NG, SS`	Jump to program line `SS` if Register `*` is not greater than the other value in Compare Unit `*`
`NOOP`	No instruction, continue to the next line
`STOP`	End the program