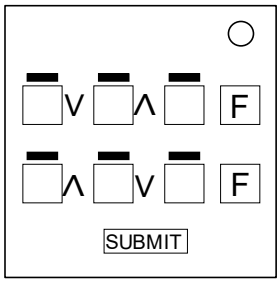
On the Subject of Logic



Logic is easy, but logic AND bomb defusal might not.

- Each row displays 3 letters. Each letter represents a statement which can be found in table 1.
- On each row, the GREEN statements should be considered as if they were enclosed in parentheses. These must be solved before the rest.
- Statements are joined with logical connective symbols. Find how each symbol works in table 2.
- Apply negation (NOT gate: false becomes true and true becomes false) to each statement first if the red LED above that statement is lit.
- Find the end result of each row, and then use the T/F button to the right to select True/False. Press "Submit" when done.

See Appendix A for indicator identification reference.
See Appendix B for battery identification reference.
See Appendix C for port identification reference.

Table 1: Statement list

Letter	Statement	Letter	Statement
A	Number of batteries = number of indicators	N	More than 2 battery holders
B	Serial number has more letters than digits	O	Has both lit and unlit indicators
C	Has IND indicator	P	Has parallel port
D	Has FRK indicator	Q	Exactly 2 ports
E	Exactly 1 unlit indicator	R	Has PS/2 port
F	More than 1 port type	S	Sum of digits in serial number > 10
G	2 batteries or more	T	Has MSA indicator
H	Less than 2 batteries	U	Exactly 1 battery holder
I	Last digit of serial number is odd	V	Serial number contains vowels
J	More than 4 batteries	W	No indicators
K	Exactly 1 lit indicator	X	Exactly 1 indicator
L	More than 2 indicators	Y	More than 5 ports
M	All ports are unique	Z	Less than 2 ports

Table 2: Logical connective symbol list

Logical Connective	Symbol	Logic Gate Equivalent	Meaning
Conjunction	∧	AND	Returns true if all inputs are true. Else returns false.
Disjunction	∨	OR	Returns true if any input is true. Else returns false.
Exclusive Disjunction	⊻	XOR	Returns true if exactly one input is true. Else returns false.
Alternative Denial	|	NAND	Returns false if all inputs are true. Else returns true.
Joint Denial	↓	NOR	Returns false if any input is true. Else returns true.
Biconditional	↔	XNOR	Returns false if exactly one input is true. Else returns true.
Implication (Left)	→	-	Returns false when left input is true and right input is false. Else returns true.
Implication (Right)	←	-	Returns false when left input is false and right input is true. Else returns true.