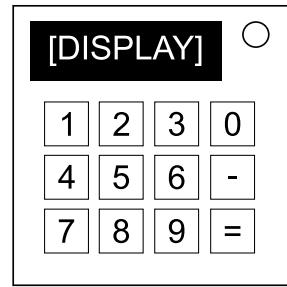


On the Subject of OMISSION

IT'S BASIC ARITHMETIC... HOW HARD COULD IT BE? =) (8 :|

An “OMISSION” module has a keypad with ten of the keys labeled from 0 through 9, another key labeled with a dash, and the twelfth key labeled with an equals sign.

The display will contain five characters. The first four characters are symbols that can be split into two text emoticons. The final character will either be a plus or minus sign.



This is a boss module.

After the solve of every non-ignored module, the display will change to another set of two emoticons followed by a plus or minus symbol. Every set of five characters is called a “stage”. The first stage is the initial display before any modules have been solved.

Order the displays by stage number and concatenate every stage into one large expression as you receive them. The final stage is indicated by a set with only emoticons.

Once all other non-ignored modules have been solved, the display turns blank and is ready to accept an answer, explained on the second page.

To show the current stage’s number, press the equals key. The stage number will be shown on the display for a second before returning to the set of five characters.

If you’ve missed any previous stages, press the dash key, type the number of a stage you previously missed, then press the equals key. The module will generate a new set of characters for that stage and every stage after it until the current stage. Until the next solve, the display will slowly cycle through the newly generated stages with a break to indicate resets.

Translate every emoticon into a digit using the table below.

This is important for determining the final solution.

0	1	2	3	4	5	6	7	8	9
:)	=()	(:)=	:()):	=)	(=	:	:
=	=	:]	=[[:] =	:[] :	=]	[=
8	8]	[8	8[]8	8)	(8	8()8	8;

As you receive each stage and add them to the expression, some specific parts of the expression must be omitted entirely. Use the following table of “omission patterns” to determine if you need to omit part of the expression upon receiving each stage.

After omitting a section from the expression, connect both sides as if nothing was there. If this causes another omission pattern to emerge, omit the new pattern as well.

Shorter omission patterns take precedence over longer ones.

“#” must be replaced by any digit.

“±” must be replaced by either a plus sign or minus sign.

#7-00±	02±#3-	#5±06-	6#-13±	15±#4+	18+#3±
22-4#±	23±7#-	#6-24±	#1±25+	28±1#-	30±#2-
32±7#-	1#±33+	34-#2±	36±#6-	8#-37±	2#±38+
#2+40±	#4±41+	#5+46±	4#-49±	53-3#±	57+0#±
62+4#±	3#-63±	66±6#-	#3+67±	0#-74±	77±9#-
78±#7+	81+#4±	2#±83+	3#+88±	5#±90+	91-#3±
9#±12±5#+		59-##±4#-		3#±6#±65+	
72±7#±4#-		3#±3#-82±		#5±4#±94-	
04±##+#0±09±		3#+0#±7#±64±		##+##-##+##-##+	

Once OMISSION is ready to accept an answer, evaluate the final expression and input the result back into the module. Pressing the dash key converts the input into a negative number. Pressing it again converts it back to a positive number. Holding the dash key resets the display.

Press the equals key to submit the input. The correct input will solve the module. An incorrect input will cause a number of strikes depending on the difference between the incorrect input and the correct evaluation. The module will then solve.

- If the input is off by a maximum of 10, you'll receive one strike.
- If the input is off by a maximum of 50, you'll receive two strikes.
- If the input is off by a maximum of 250, you'll receive three strikes.
- If the input is off by more than 250, you'll receive four strikes.