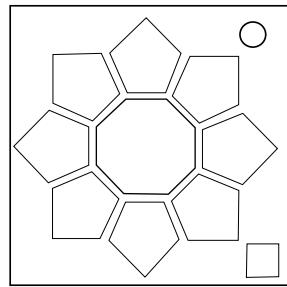


On the Subject of Simon Stores

Calculators at the ready...

This module consists of nine buttons, six of which have the colours: (R)ed, (G)reen, (B)lue, (C)yan, (M)agenta, and (Y)ellow.



These buttons will flash in an increasing sequence, starting with the centre button, in which each flashing colour corresponds to an operator in the tables below.

Apply each operation successively until the end of the sequence and use the result to determine the sequence of coloured buttons to press for each stage.

If at any point an operator yields-

- a value greater than 364, then subtract 365 from the value until it is less than 364.
- a value less than -364, then add 365 to the value until it is greater than -364.

Section 1: Determining Initial Values

To determine the number to be entered into the sequence of operators, interpret the following pairs of digits in the serial number as base36 numbers modulo 365 (the example SN = 'B01MB2' is used to illustrate how this is done):

- For stage 1, use the 3rd and 4th digits. ($a_0 = '1M' = (1 * 36 + 22) \bmod 365 = 58 \bmod 365 = 58$)
- For stage 2, use the 5th and 6th digits. ($b_0 = 'B2' = (11 * 36 + 2) \bmod 365 = 398 \bmod 365 = 33$)
- For stage 3, use the 1st and 2nd digits. ($c_0 = 'B0' = (11 * 36 + 24) \bmod 365 = 420 \bmod 365 = 55$)

Section 2: Operation Tables

Use the tables below to calculate each successive value of the sequence.

- n is the current step of the sequence.
- D is the sum of the base36 digits of the serial number. ('B01MB2' = 11 + 24 + 1 + 22 + 11 + 2 = 69)
- The operators listed are of the form $x_n = F(x_{n-1})$, where F is one of the operators below.

Table S1: Single Colour Flashes

	Stage 1	Stage 2	Stage 3
R	$a_n = a_{n-1} + D$	$b_n = b_{n-1} + a_{n-1} + n^2$	$c_n = c_{n-1} + b_{n-1} - a_{n-1}$
G	$a_n = a_{n-1} - D$	$b_n = 2b_{n-1} - a_{n-1}$	$c_n = c_{n-1} - 2b_{n-1}$
B	$a_n = 2a_{n-1} - D$	$b_n = 2b_{n-1} - a_0 - 4n^2$	$c_n = c_{n-1} + b_0 - a_3$
C	$a_n = D - a_{n-1} - 8n$	$b_n = b_{n-1} + a_1$	$c_n = c_{n-1} - b_{n-1} + a_{n-1}$
M	$a_n = 3n^3 - 2a_{n-1}$	$b_n = b_{n-1} + a_2 - D$	$c_n = c_{n-1} - 2a_{n-1}$
Y	$a_n = a_{n-1} + D - 6n$	$b_n = b_{n-1} + a_3 - a_{n-1}$	$c_n = c_{n-1} + b_4 - a_0$

Table SS2: Double Colour Flashes

The operators here depend on whether the flashing colours are (P)primary or (S)econdary with respect to additive colour mixing.

P and S refer to the operators in table SS1 that are associated with the individual flashing colours.

	Stage 1	Stage 2	Stage 3
2P	$a_n = \max(P_1(a_{n-1}), P_2(a_{n-1}))$	$b_n = \text{abs}(P_1(b_{n-1}) - P_2(b_{n-1}))$	$c_n = P_3(c_{n-1}) + P_3(b_{n-1}) + P_3(a_{n-1})$
1P, 1S	$a_n = P(a_{n-1}) + S(a_{n-1}) - 2D$	$b_n = 4D - \text{abs}(P(b_{n-1}) - S(b_{n-1}))$	$c_n = \min(P(c_{n-1}), S(c_{n-1}), -\text{abs}(P(c_{n-1}) - S(c_{n-1})))$
2S	$a_n = \min(S_1(a_{n-1}), S_2(a_{n-1}))$	$b_n = \max(S_3(b_{n-1}), S_3(a_{n-1}))$	$c_n = S_3(c_{n-1}) - S_1(c_{n-1}) - S_2(c_{n-1})$

P_3 and S_3 refer to the operators that correspond to the respective primary and secondary colours that did not flash at this step of the sequence.

Table SS3: Triple Colour Flashes

	Stage 1	Stage 2	Stage 3
3P	$a_n = a_{n-1} + a_0$	$b_n = b_{n-1} + (b_{n-1} \bmod 4)b_0 - a_3$	$c_n = c_{n-1} + (c_{n-1} \bmod 3)c_0 - (b_{n-1} \bmod 3)b_0 + (a_{n-1} \bmod 3)a_0$
2P, 1S	$a_n = \max(P_1(a_{n-1}), P_2(a_{n-1}), S(a_{n-1}))$	$b_n = b_{n-1} + P_1(b_{n-1}) + P_2(b_{n-1}) - S(a_{n-1})$	$c_n = P_1(c_{n-1}) + P_2(c_{n-1}) - S(b_{n-1}) - S(a_{n-1})$
1P, 2S	$a_n = \min(S_1(a_{n-1}), S_2(a_{n-1}), P(a_{n-1}))$	$b_n = b_{n-1} + S_1(a_{n-1}) + S_2(a_{n-1}) - P(b_{n-1})$	$c_n = S_1(c_{n-1}) + S_2(c_{n-1}) - P(b_{n-1}) - P(a_{n-1})$
3S	$a_n = a_{n-1} - a_0$	$b_n = b_{n-1} + (b_0 \bmod 4)b_{n-1} - a_3$	$c_n = c_{n-1} + (c_0 \bmod 3)c_{n-1} - (b_0 \bmod 3)b_{n-1} + (a_0 \bmod 3)a_{n-1}$

Note: a_4 is equal to zero.

Section 3: Converting Results

Once a result of the sequence of operations has been obtained, it must be converted into **balanced ternary**.

Balanced ternary is a base 3 number system in which the digits have the values 1, 0, and -1.

For example: $250 = 243 + 9 - 3 + 1 = 3^5 + 3^2 - 3^1 + 3^0 = +00+-+$

The balanced ternary representations of integers from -10 to 10 are shown below.

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
-0-	-00	-0+	-+-	-+0	-++	--	-0	-+	-	0	+	+-	+0	++	---	+0-	---	+0-	+00	+0+

Section 4: Entry and Submission

To work out which buttons need to be pressed, first apply the list of conditions on the positions of buttons on the module to each row of the initial correspondences table, starting from the top of the list and working down.

Initial Correspondences	3^0	3^1	3^2	3^3	3^4	3^5
Stage 1	R	G	B	C	M	Y
Stage 2	Y	B	G	M	C	R
Stage 3	B	M	R	Y	G	C

If...	...,then...
the top right button is yellow	cycle each element one space to the right. (1->2, 2->3,...,6->1)
the red button is opposite the cyan button	swap each element with its complementary colour. (R<->C, G<->M, B<->Y)
the green button is adjacent to the white button	cycle the primary colour elements. (R->G, G->B, B->R)
the magenta button is adjacent to the black button	cycle the secondary colour elements. (C->M, M->Y, Y->C)
the blue and yellow buttons are on the same side of the module	swap B with the element opposite. (1<->6, 2<->5, 3<->4)
the red button is on the right side of the module	swap R and Y.
the blue button is on the left side of the module	swap G and C.

Once a balanced ternary number and the buttons corresponding to powers of 3 for the current stage have both been determined, follow the instructions below:

1. Press the centre button. This will cause the sequence to stop flashing and light the white button.
2. Using the coloured buttons, enter the non-zero balanced ternary digits in order of least to most significance:
 - Pressing the white and black buttons will toggle which of the two are lit.
 - Pressing a coloured button while the white button is lit will enter a '+' in the corresponding digit.
 - Pressing a coloured button while the black button is lit will enter a '-' in the corresponding digit.
3. Press the centre button again to submit the sequence of inputs.

Submitting the correct sequence of inputs will advance the module to the next stage.

Submitting an incorrect balanced ternary number, or submitting the coloured buttons in the wrong order will result in a strike.