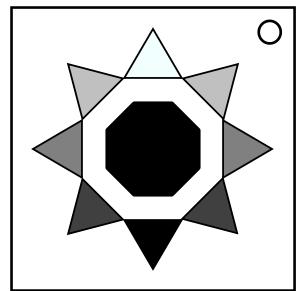


## On the Subject of Deceptive Rainbow Arrows

*What do you mean, "Point at Inverted Stores' Calculations!?" It points to the void and two useless bits!*



This module has eight arrow buttons in a randomized rainbow color arrangement, and a display screen in the center showing a flashing number from 00 to 77. The eight possible colors the module can show are red, orange, yellow, green, blue, purple/violet, white, and black. When not submitting, there are 6 sets of flashing arrows followed by the text flashing a different color, denoting the end of the arrow flashing sequence.

**If the expert is not prepared to do this module, tell the defuser to discard this module immediately. Do not continue looking at this manual. Do not ask the defuser about it. Just. Throw it away. Burn the manual if necessary.**

To solve the module, the defuser will need to query a series of 2 colors and read out the response shown from the module. The response shown will be in base-8 or octal. Go through the procedure for each response to get the next sequence of colors to query. Once the defuser has queried for the third response, determine the value to submit based on the response of that query.

To query on the module, press the two arrow colors and then the display to query for the response. The response will be shown in "R:##" where ## is the octal representation. To submit on the module, press the display while no arrows are pressed, press the two arrow colors in order, and then the display again to submit. If the defuser wants to exit out of submission, press the display while no arrows have been inputted. **Do not query or submit with 1 color pressed, or interact while the module is processing the query. This will cause a strike and display an error message on the module for a few seconds before reverting back to its initial state.**

## Contents on This Manual:

- [The First Query](#)
- [The Second Query](#)
- [The Third Query](#)
- [Submission The Response](#)

## The First Query

To obtain the first query, create 3 pairs of values from the serial number; by taking the 1st and 2nd, 3rd and 4th, 5th and 6th, all as separate pairs; converting letters into their A1Z26 counterpart. Take the sum of each pair and modulo them all by 8 to obtain 3 separate values. These three values are referred as Z, Y, and X respectively.

Find the row representing Z in the table provided to get a series of colors that represent the octal values 0-8 in that order. This row will be used for later queries.

If an empty port plate is present, the color representing Y will be the first color in the pair to query and the color representing X will be the second color in the pair to query.

Otherwise, the color representing X will be the first color in the pair to query and the color representing Y will be the second color in the pair to query.

For each cell in the table, each letter is represented by their first letter in the color name, except black which is represented with "K" to avoid confusion with blue.

Z	Octal Digit Represented By Color							
	0	1	2	3	4	5	6	7
0	R	K	Y	W	O	G	P	B
1	Y	O	B	G	R	K	W	P
2	B	Y	K	R	W	P	G	O
3	G	R	P	O	K	W	B	Y
4	P	B	G	K	Y	R	O	W
5	K	W	O	B	P	Y	R	G
6	W	G	R	P	B	O	Y	K
7	O	P	W	Y	G	B	K	R

## The Second Query

Convert the flashing arrow sequence into a string of 6 binary digits, with the first binary digit represented by the first set of flashing arrows after the text has flashed. White and black will NEVER flash in the arrow sequence, and should NOT be considered for deciding if the flashing set is inverted or not. If all but 1 colored arrow is flashing, that binary digit is represented as a 1. Otherwise, it is represented as a 0.

Convert your response from the first query from octal to binary, using the table provided. Create this as another binary string.

Oct	Bin																
0	000	1	001	2	010	3	011	4	100	5	101	6	110	7	111		

XOR the two binary strings obtained earlier to get a new binary sequence. Take every second binary digit from that sequence to get 1 binary triplet and create the second with the remainder. Convert these two binary triplets into their octal digits, join the octal digits into one octal value, and then cross reference this table to get a new octal value. Query the new octal value with the color arrangement determined for the first query.

	-0	-1	-2	-3	-4	-5	-6	-7
0_-	36	02	70	74	57	32	24	22
1_-	40	30	06	17	64	34	51	47
2_-	63	53	45	54	10	61	60	41
3_-	67	07	00	42	04	56	46	65
4_-	16	62	77	26	31	12	66	25
5_-	55	52	11	14	27	05	35	76
6_-	15	50	37	23	44	33	43	73
7_-	75	72	03	21	20	13	71	01

## The Third Query

Use the response that was obtained from the second query and start on the coordinate in reading order based on the grid provided, where 00 is the top-left of the grid. On the grid, start facing in the same direction as the white arrow on the module. Go through this snippet of psudo-code:

1. Let I = 0, P be an empty list, and M the number of arrows clockwise from the white arrow to the arrow matching the color of the text when the text flashes.
2. While I < 4, repeat the following nested steps:
  1. Let E be a separate list. and A = 0 in this section.
  2. While A < I + 1, repeat the following nested steps:
    1. Append the value you are on to E.
    2. Move 1 step in the current direction, wrapping around to the other side of the grid if moving to out of bounds.
    3. Add 1 to A.
  3. Append the sum of all the values in E, modulo 8, on to P.
  4. Rotate 45° clockwise M times.
  5. Add 1 to I.
3. Take P[0] and XOR it with P[1] to create A in octal, where P[0] is the first element in list P. Likewise, take P[2] and XOR it with P[3] to create B in octal.
4. Find the cell corresponding to the 2 digit concatenation of A, B from the second query's table and query the respecting color combination from the first query's row.

0	1	7	3	5	4	2	6
1	3	0	4	7	6	5	2
3	4	1	6	0	2	7	5
4	6	3	2	1	5	0	7
6	2	4	5	3	7	1	0
2	5	6	7	4	0	3	1
5	7	2	0	6	1	4	3
7	0	5	1	2	3	6	4

## Submitting The Response

To submit the response, use the initially displayed value, all responses from the queries you have up to this point, and the 6 sets of arrow flashes that have occurred. Examine the ports present, and omit the flashes whose position is the most amount of the ports of that type listed in this table. If there is a tie for the highest amount of ports, omit multiple positions that are tied to their highest occurring port type, unless there is a 4, 5, or 6-way tie, omit the 1st, 3rd, and 5th flashes in that set.

Port Type	Parallel	Serial	PS/2	RJ-45	Stereo RCA	DVI-D
Position To Omit	2nd	6th	3rd	4th	5th	1st

Go through the rest of the flashing sets and find its respective equation based on the color it flashed. Solve for  $x$  if the flash is inverted or  $F(x)$  otherwise by plugging on the opposing variable (solving for  $F(x)$ , plug in  $x$ , and vice versa, after plugging in the other values), the last value that was used to calculate the set of flashes, or the displayed value if this is the first set. Keep the calculated value within 0 to 63 by adding or subtracting 64 repeatedly if the calculated value in base 10 after the current flash set is not. Omit decimals when dividing, I.E. make 24.5555 become 24, NOT 25. On the table provided:

Color Flashed	Equation
Red	$R(x) = x + A_0$
Orange	$O(x) + A_1 = x$
Yellow	$Y(x) + (2^n - 1) = A_2 - x$
Green	$G(x) + x = 2 * x + \sum A / 3$
Blue	$x - 2 * B(x) - 2 * n = \min(A) - B(x)$
Purple	$3 * P(x) - 2 * x = \max(A) - x + 4 * P(x)$

- $A$  is the list of responses obtained up to this point.
- $A_n$  refers to the response that was queried  $(n + 1)$ th.
- $n$  is the position of the current remaining flashing set when the equation is used, with the first set being 0.
- $\min(A)$  refers to the minimum value from list  $A$ .
- $\max(A)$  refers to the maximum value from list  $A$ .
- $\sum A$  refers to the sum of all of the values from list  $A$ .

Use the octal representation of the result of the last calculated flashing set to reference the second query's table to get a new value. Use the color representation of the new value to submit the response.