

## On the Subject of Seven Choose Four



*What... is your favourite colour? Blue. No, yellow0000000W!!*

This module consists of four pink buttons, each with a flashing color next to them.

### Tips for Success

- In order to solve this module, you must submit four 4-digit codes in sequence. Each 4-digit code you enter has a sequence number. The first 4-digit code you enter has a sequence number of 1, the second has a sequence number of 2, etc.
- Each button from top to bottom has a color flashing next to it.
- These flashing lights can be decoded into 4 "keys", each of which is a letter.
- Each flashing light follows one of four patterns: Morse Code, Tap Code, a sequence of colors, and a number of long flashes between 1 and 7.
- The morse code and tap codes are deciphered in the usual way (see morse code or tap code manuals for help). These will provide you with the **Morse Key** and **Tap Key**.
- The sequence of colors corresponds to the 4 colors associated with the buttons, in order, from top to bottom, and a fifth color, which is the **Color Key**. Use the first letter of the name of this color as the **Color Key**.
- The last code will simply be a number of flashes between 1 and 7. These flashes are longer than the morse code dash length to help differentiate between the two. The **Number Key** is the letter in the corresponding position of this list R G B M C Y W.
- Now that you have the four keys, you will use these keys as inputs to decrypt the 4-digit codes from each color.
- You only need to determine the 4-digit codes for the colors which correspond to each button.
- The order you need to submit the codes is the same as the order of the colors from top to bottom, or the first four colors of the color sequence, which is equivalent.
- The key for the first color is the **Color Key**. The order of the keys for the remaining colors continues from the location of the **Color Key** downward, wrapping around as necessary.
- The instructions to determine the 4-digit code for each color can be found below.

- When you are ready to submit the codes, note the following things:
  - When you enter a 4-digit code, treat the buttons as numbered 1-4 starting from the button associated with the current color and proceeding downward, and wrapping around as necessary.
  - The module will only issue a strike after the 4-digit code is submitted. If you receive a strike, the current color code will be deleted and you need to enter it again.
  - When a 4-digit code is entered correctly, the button associated with that color will turn green. Once all four buttons are green the module is solved.

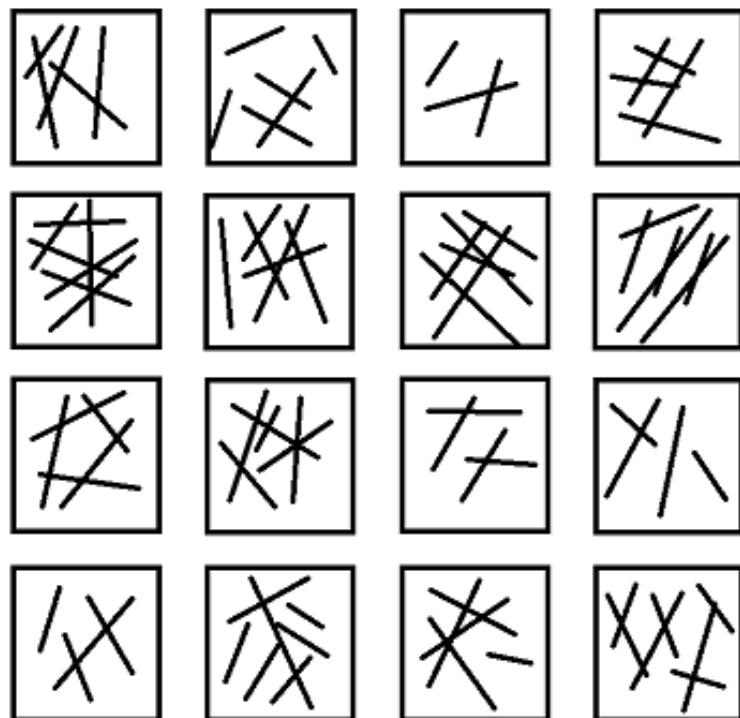
## Red

1. Convert the input key to a 2-digit number using the alphanumeric position of the key with A = 1, etc.
2. If the number is less than 10, keep the leading 0. (01, 02, etc.)
3. Add to this number the number of ports on the bomb which contain the letter R in their name. If the number is now greater than 99, use the 2 least significant digits.
4. Convert each of the two digits to 1 if they are odd, and 0 if they are even.
5. Consult the table of logic gates to determine which gate you should use, where the sequence number corresponds with which 4-digit code you are currently calculating.
6. Use the two digits as the input for the gate, with 1 being true and 0 being false (See Appendix A for help with logic gates).
7. Now, find your code in the table based on whether the output of the logic gate.

	Sequence Number			
	1	2	3	4
Logic Output	NOR	XOR	OR	AND
True	1423	2314	4123	2341
False	2313	4231	2233	1231

**Green**

1. Convert the input key to a 2-digit number using the alphanumeric position of the key with A = 1, etc.
2. If the number is less than 10, keep the leading 0. (01, 02, etc.)
3. Add to each digit independently the number of G's and 7's in the serial number, and then take each digit modulo 4.
4. Use these two digits in all four combinations as row and column (top-left is 0,0) to find four cells in the **Lines Table** (example AB => AA, AB, BA, BB).
5. Add up the number of lines in all four cells, and use this information to find the 4-digit combination in the table below.
6. Take the combination associated with the current sequence number. If the number of lines you counted is even, reverse the sequence.

**Lines Table**

Sequence Number			
1	2	3	4
3241	1122	4332	1231

**Blue**

1. Convert the input key to its alphanumeric position (A=1,B=2,etc.).
2. Subtract 5 from the value until it is within the range 1-5. This is the row value.
3. For the column, first check the serial number for first letter that appears between A and E. If no such letter exists, then convert the row number to a letter (A=1,B=2,etc.) and use this letter as the column.
4. This is your starting coordinate in the grid below.
5. First, point in the direction which allows you to travel the longest possible distance before hitting the edge of the grid. Ties are broken in favor of North and Clockwise.
6. Travel in this direction a number of spaces equal to the number of AA batteries on the bomb.
7. Then, reorient yourself again following the rules from step 5 using South-Clockwise for ties, and travel a number of spaces equal to the number of D batteries on the bomb.
8. The space you land on has the 4-digit combination.

	A	B	C	D	E
1	1234	1243	2134	2143	3124
2	3142	1324	1342	1423	1432
3	4123	4132	2314	2341	3214
4	3241	2413	2431	4213	4231
5	3412	3421	4312	4321	1234

**Magenta**

1. Find the key letter in the grid below.
2. Also locate the first four characters of the serial number in the grid, converting numbers to letters (A=0,B=1,etc.).
3. Calculate the number of spaces it would take to travel the shortest path, only moving orthogonally (not diagonally), to each of the four characters noted from the serial.
4. Restart from the key letter each time, and subtract or add 4 to each distance to make sure it stays within the range 1-4.
5. These four modified distances in order, are the 4-digit combination.
6. A and Z are interchangeable in the grid.

L	P	R	I	S
X	O	F	B	V
A	Y	D	C	T
K	G	W	Q	N
M	E	U	H	J

Cyan

1. Convert the key to a number (A=1,B=2,etc.) and subtract 7 until the number is between 1 and 7.
2. This number corresponds with the color in the corresponding position of this list **R G B M C Y W**.
3. You need 3 other colors to find the combination, so you will use the 3 other colors corresponding to the other 3 buttons on the module.
4. Find your combination by concatenating the 4 values you obtain based on the number of primary and secondary colors among these 4 colors. The two primary numbers should be placed before the two secondary numbers (**XXYY**).
5. For the purposes of calculating **P** and **S**, the number of primary and secondary colors, respectively, **R G B** and **W** count as primary, and **M C Y** and **W** count as secondary.

#P	<b>XX</b>	#S	<b>YY</b>
<b>1P</b>	13	<b>4S</b>	14
<b>3P</b>	42	<b>0S</b>	34
<b>4P</b>	43	<b>1S</b>	23
<b>0P</b>	12	<b>3S</b>	24
<b>2P</b>	13	<b>2S</b>	31

## Yellow

1. To find the 4-digit combination, we will use a special encryption on the word **YELLOW**, which will depend on the key.
2. First remove the letter from the word **YELLOW** which is closest to the key letter. Ties are broken in favor of the letter further in the alphabet. If the letter to remove is **L** only remove one of them.
3. Now find the inverse letter of the key (**A->Z,B->Y,C->X,etc.**) and repeat step 2.
4. You should have 4 letters left in the word **YELLOW**. Shift these letters to the right based on the sequence number, wrapping the letters around as needed.
5. Now replace the letters with numbers 1-4 based on the order in which they appear in the alphabet (Ex. **AFDH = 1324**). This is your solution.

## White

- Using the key, and the current sequence number, choose any of the other 6 colors, and any of the 4-digit combinations they produce will work as the solution for this color.
- **NOTE:** Since you are using a new key and this 4-digit code corresponds to a different sequence number, it is highly unlikely that the 4-digit code produced by a color will be the same, so entering the same code you already entered for another color on the module likely will not work!

## Appendix A

Decimal	Binary	Logic Gates			
		NOR	XOR	OR	AND
0	00	1	0	0	0
1	01	0	1	1	0
2	10	0	1	1	0
3	11	0	0	1	1