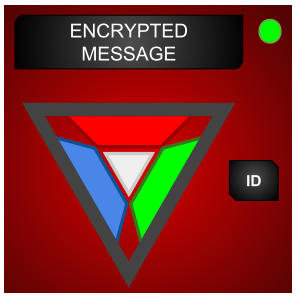
On the Subject of Unfair Cipher



It doesn't play that fair anymore.

This module has two displays. The display on top shows the encrypted message.

The display on the right can be clicked to toggle between showing the Module ID, in white, or strikes the module keeping track of, in red.

- *Both of these are shown in Roman numerals!*

The module encrypts a string of ten three letter long instructions with two **Playfair Ciphers**, using different **keys** for each. Enter the correct combination of inputs to disarm the module.

- The order of encryptions is the following: **Original -> Key A Encrypted -> Key C Encrypted -> Caesar Ciphered.** Reverse the order to obtain the **original** instruction string.

Key A

1. Start with the bomb's serial number.
2. Transform each letter into its numerical equivalent (A=1, B=2, etc.).
    - Make a single string of digits.
    - Ignore the first character if its numerical equivalent is 20 or above.
3. Remove the last digit if either the 4th or the 5th characters of the serial are **vowels.**
    - You should only do this once, even in the case both characters are vowels.
4. Convert this number into **hexadecimal**, refer to *Appendix D3K2H3X* for instructions.
5. If the hexadecimal number has two digits in range 1 to 26 next to each other, transform them into the alphabetical equivalent. Otherwise, transform the single digits into their alphabetical equivalents.
    - Delete all zeroes which don't have either 1 or 2 on their left.
        - **(Ex.: 0129A -> LIA)**
        - **(1029A -> JBIA)**
6. Transform the Module ID, the amount of Port Plates and the amount of Battery Holders into their alphabetical equivalents.
7. Append these three characters at the end of the result of the previous conversion.
8. **This is Key A.**

Key B

Key B is needed to obtain **Key C** from **Key A.**

Obtain your initial Key B from the following table:

		Month (Jan – Jun)					
		Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**
Day	**Mon**	AB41	6522	DB83	B124	DB95	AFE6
	Tue	AB42	6523	DB84	B125	DB96	AFE7
	Wed	AB43	6524	DB85	B126	DB97	AFE8
	Thu	AB44	6525	DB86	B127	DB98	AFE9
	Fri	AB45	6526	DB87	B128	DB99	AFEA
	Sat	AB46	6527	DB88	B129	DB9A	AFEB
	Sun	AB47	6528	DB89	B12A	DB9B	AFEC

		Month (Jul – Dec)					
		Jul	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
Day	**Mon**	AFC7	C178	D5A9	FE1A	EFAB	453C
	Tue	AFC8	C179	D5AA	FE1B	EFAC	453D
	Wed	AFC9	C17A	D5AB	FE1C	EFAD	453E
	Thu	AFCA	C17B	D5AC	FE1D	EFAE	453F
	Fri	AFCB	C17C	D5AD	FE1E	EFAF	4540
	Sat	AFCC	C17D	D5AE	FE1F	EFB0	4541
	Sun	AFCD	C17E	D5AF	FE20	EFB1	4542

If your initial Key B has numbers, transform them in their **alphabetical equivalent,** using the **same** rules applied for Key A.

Key C

To obtain Key C, Playfair encipher Key A using Key B as a **Keyword.**

Refer to *Appendix PL4YF4112 101* for instructions.

The result of such encryption is **Key C.**

Read onwards only once you have all three keys, which should be ALPHABETICAL characters.

You're all set to decipher the instructions. The first cipher to solve is a simple Caesar Cipher.

Solving – Step 1: Caesar Cipher

To calculate the offset used for the Caesar Cipher use the following table:

	Offset + Add – Subtract
For every port **type**	– 2
For every **port plate**	+ 1
For every **consonant** in the serial number	+ 1
For every **vowel** in the serial number	– 2
For every **on** indicator	+ 2
For every **off** indicator	– 2
For every **battery**	– 1

Finalize your offset by checking if any of these conditions apply. (Always drop any remainder.)

Condition	Operation
No batteries	– 10
No ports	× 2
31 or more modules	÷ 2

This is the offset used in the Caesar Cipher. To decipher it, shift every letter on the screen by **X** letters forwards if the offset is negative, backwards if positive.

Wrap back to the other side of the alphabet if you have to go backwards from **A** or forwards from **Z**.

Solving – Step 2: Playfairs

Playfair decrypt the string you just deciphered using Key C as a keyword.

Repeat this process using Key A as a keyword.

You now have the original message, congratulations.

Solving – Step 3: Executing the Instructions

- The message consists of 10 instructions, each with potentially different inputs required. Execute the instructions left to right.
- Unlike Playfair Cipher, getting a strike anywhere on the bomb won't change the answer you worked VERY hard on.
  - However, striking anywhere on the bomb **WILL** affect some inputs.
- Tap the small screen on the right to toggle between showing Module ID or Strikes. Strikes are shown in red. Both of these are in roman numerals.
  - The module counts strikes coming from EVERY module on the bomb, not only itself.
  - The module can keep track of strikes even in Time Mode!

Instructions:

'%' refers to the modulo (remainder) operation.

- PCR: Press the Red button.
- PCG: Press the Green button
- PGB: Press the Blue button
- SUB or MIT: Press Inner/Outer center...
  - If SUB: Press **Outer Center** when the seconds digits on the timer match.
  - If MIT: Press **Inner Center** when the seconds on the timer match $(m + c) \% 60$, with m being the Module ID, and c being the number a colored (R, G, B) button has been pressed since the last strike on this module.
    - "MIT" will accept inputs up to 2 seconds early or late
- PRN or CHK:
  - If PRN: Press **Inner Center** if Module ID % 20 is a prime number; Otherwise, press Outer Center.
  - If CHK: Press **Outer Center** if Module ID % 20 is a prime number; Otherwise, press Inner Center.
  - **Refer to *Appendix PRIM3* for a list of prime numbers.**
- BOB: Press **Inner Center**
  - If there is a lit BOB as the only indicator on the bomb and two batteries in total, this **instantly solves the module!**
- REP or EAT: Repeat the last input.
  - If this comes out as first instruction, press **Inner Center**
- STR or IKE: Starting from Red (0), the colored button on top, count as many colored buttons clockwise as there are strikes and press the resulting button.

Appendix - D3K2H3X

Follow these steps to convert any integer into an hexadecimal number:

1. Divide the decimal number by 16
2. Keep the quotient of such division aside for the next iteration.
3. Convert the integer remainder of such division into the HEX digit using the table below.
4. Repeat steps 1 to 3 until the quotient is zero.
5. The HEX number is read backwards, starting from the "quotient zero" operation.
    - Neglect the zero if it's the first digit.

DEC	HEX
0 - 9	0 - 9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
26	1A
...	...

Appendix - PL4YF4112 101

- Create a a 5×5 matrix of letters. Start with your **keyword** and fill the rest with the unused letters of the alphabet. Each letter must occur only **once** in the matrix, so only add the first occurrence. 'J' and 'I' are interchangeable.
- Split the **message** into character pairs. If you cannot form a pair, add an **'X'**. For each pair:

If DECRYPTING:

- If the letters appear on the same row of your matrix, replace them with the letters to their immediate left respectively, wrapping around to the right side of the row if necessary.
- If the letters are on the same column of your matrix, replace them with the letters immediately above, wrapping to the bottom if necessary.
- If the letters are on different rows and columns, replace each of them with the letter on the same row but in the column of the other letter in the original pair.

If ENCRYPTING:

- If the letters appear on the same row of your matrix, replace them with the letters to their immediate right respectively, wrapping around to the left side of the row if necessary.
- If the letters are on the same column of your matrix, replace them with the letters immediately below, wrapping to the top if necessary.
- If the letters are on different rows and columns, replace each of them with the letter on the same row but in the column of the other letter in the original pair.

- Drop any X's that don't make sense and locate any I's that should be J's.

Appendix - PR1M3

- Prime numbers (to 20): 2, 3, 5, 7, 11, 13, 17, 19