|  |  |  |
| --- | --- | --- |
| |  | | --- | | Dec 8, 2016 at 4:38pm  [Quote](https://cicada3301.boards.net/post/64/quote/41)[like](https://cicada3301.boards.net/thread/41/pages-49-256-byte-strings)  Post Options  **Post by mortlach on Dec 8, 2016 at 4:38pm**  **Pages 49 to 51 and 256 Byte Strings**  Pages 49 to 51 contain the alpha-numeric string given at the bottom of this post. In the string there are 256 pairs. The first character of each pair is always a 0,1,2,3 or 4, and the second is one of 59 (prime) unique characters, the digits 0 to 9, 26 upper case letters and 23 lower case letters:      0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x  **Another Skipped f**  It’s interesting to note that 'f' has been skipped, something similar is found when solving the earlier pages (in the pages that have a decryption key each letter f in the message **is not encrypted**, f is used as an [*interrupter*](http://cicada3301.boards.net/thread/38/decrypt-runes-vigen-re-cipher)*,*pages that are just *encoded* do not skip 'f').  **Interpreting The String**   One way to interpret this string is that it is a representation of numbers in Base64, Base 60, Base 59, or something else?  Typical [Base64 encoding](https://en.wikipedia.org/wiki/Base64) starts with capital letters, then lower case letters, then numbers:      A = 0, B = 1, C = 2, …     a = 26, b = 27, c = 28 …     0 = 52, 1 = 53, 2 = 54 …   After some discussion on solvers we decided to try the Base 59 mapping of characters to numbers. We chose to start with the numbers, then the 26 Upper case letters and then the 23 lower case letters, giving [this map from each possible base 59 pair to decimal](http://pastebin.com/KH4sH6Se). There are other choices but this seems reasonable (and gives us an interesting result).  **A 256 Byte String**  Using the Base 59 assumption [we get a decimal number for each pair](http://pastebin.com/b2D5fAxJ).  All the decimal numbers are in the range 0 to 255, an unsigned 8 bit (1 byte) number, (“unsigned char” in c). What makes this more interesting is that in the 2014 we found three 256 byte numbers that were never used:  [pastebin.com/raw/qePehdKM](http://pastebin.com/raw/qePehdKM) [uncovering-cicada.wikia.com/wiki/CICADA\_3301\_2014\_PUZZLE\_FACTS\_PART\_5](http://uncovering-cicada.wikia.com/wiki/CICADA_3301_2014_PUZZLE_FACTS_PART_5)  **Four 256 Byte Strings**  We now have four 256 byte strings. At first glance the 4 strings look random, perhaps they can be combined in someway, or they are encrypted, or they are a key or a passphrase that will help us make progress?  \*Comments, questions, suggestions, omissions etc ? please try [#cicadasolvers](http://webchat.freenode.net/?channels=cicadasolvers)  MSGA  **Page 49 to 51 String:**   3N 3p 2l 36 1b 3v 26 33  1W 49 2a 3g 47 04 33 3W  21 3M 0F 0X 1g 2H 0x 1R  1n 3I 2r 0P 2U 16 2L 2D  1t 1s 3H 0d 0s 1K 2D 05  1K 1O 0S 1D 3o 1l 3J 1G  4D 0G 0l 0x 1Q 2p 2a 1K  4E 1w 2Q 19 1k 3G 24 0p  22 4F 0P 3C 3J 1D 2n 1m  2i 1J 3P 2v 1s 2O 0k 1M  2M 0w 3L 3D 2r 0S 1p 15  3V 3e 3I 0n 3u 1O 0u 0Z  3g 2U 1C 0Y 1N 3n 0W 3Q  22 13 0V 3c 0E 34 0W 1t  1D 2N 3H 47 0s 2p 0Z 34  0g 3v 1Q 0s 0D 0K 2h 3D  3L 2x 1Q 20 2n 2L 1C 2p  0A 29 3r 0D 45 0k 2e 2W  25 3U 1W 2r 46 2s 2X 39  3p 0X 0E 1q 0q 4B 49 48  3r 3b 3C 1M 1j 0I 4A 48  40 3m 4E 0s 2S 1v 3T 0I  3t 2B 2k 2t 2O 0e 2l 1L  28 2a 0J 1L 0c 3C 2o 0X  00 2Z 2d 1T 2u 1t 1j 0l  1o 1E 3T 18 3E 1G 27 0L  0v 2t 06 11 1A 2U 4B 1O  2M 3d 2S 0x 0w 0q 0p 2V  18 0q 1D 49 2O 00 1v 2t  1k 3s 3G 21 3w 0W 29 2r  2O 2L 0g 3Y 0M 0u 3I 3C  1r 2c 2q 3o 30 0a 39 1K | | *Last Edit:* Dec 10, 2016 at 1:12pm by [mortlach](https://cicada3301.boards.net/user/3) | |
| |  |  | | --- | --- | | onecool Guest  Guest Avatar | Dec 12, 2016 at 9:33am  [Quote](https://cicada3301.boards.net/post/68/quote/41)[like](https://cicada3301.boards.net/thread/41/pages-49-256-byte-strings)  Post Options  **Post by onecool on Dec 12, 2016 at 9:33am**  200,227,164,183,96,233,124,180,91,245,154,218,243,4,180,209,119,199,15,33,100,135,58,86,107,195,170,25,148,65,139,131,113,112,194,39,53,79,131,5,79,83,28,72,226,105,196,75,249,16,46,58,85,168,154,79,250,116,144,68,104,193,122,50,120,251,25,189,196,72,166,106,161,78,202,174,112,142,45,81,140,57,198,190,170,28,109,64,208,217,195,48,232,83,55,35,218,148,71,34,82,225,32,203,120,62,31,215,14,181,32,113,72,141,194,243,53,168,35,181,41,233,85,53,13,20,160,190,198,176,85,118,166,139,71,168,10,127,229,13,241,45,158,150,123,207,91,170,242,171,151,186,227,33,14,110,51,247,245,244,229,214,189,81,103,18,246,244,236,224,250,53,146,115,206,18,231,129,163,172,142,40,164,80,126,154,19,80,38,189,167,33,0,153,157,88,173,113,103,46,108,73,206,67,191,75,125,21,56,172,6,60,69,148,247,83,140,216,146,58,57,51,50,149,67,51,72,245,142,0,115,172,104,230,193,119,234,32,127,170,142,139,41,211,22,55,195,189,111,156,169,226,177,36,186,79  as comma separated list of decimals ^ | |  | |
| |  |  | | --- | --- | | Lurker69 Guest  Guest Avatar | Jan 12, 2017 at 1:18am  [Quote](https://cicada3301.boards.net/post/72/quote/41)[like](https://cicada3301.boards.net/thread/41/pages-49-256-byte-strings)  Post Options  **Post by Lurker69 on Jan 12, 2017 at 1:18am**  I am not completely sure which one is correct so I am posting both version of 4th stringpastebin.com/kmn59PAi](http://pastebin.com/kmn59PAi) [github.com/rtkd/iddqd/blob/master/byte-strings/byte-strings](https://github.com/rtkd/iddqd/blob/master/byte-strings/byte-strings)  <crashdemons> [pastebin.com/miEZxYE1](http://pastebin.com/miEZxYE1) <crashdemons> I think that's the right one <crashdemons> for base59 and base61 conversions <crashdemons> base62 was fail: one of the "bytes" had a value over FF <crashdemons> for anyone interested in reproducing it, Radix-N (arbitary) to base 10 (decimal) is done by sum( Vn\*B^Pn ) for n=0 to the word length across the Base-B digits where B is the base, Vn is the decimal digit value (position in the alphabet of the Base), and Pn is the 0-based position in the 'word' <crashdemons> so Pos('3')\*59^1 + Pos('N')\*59^0 etc <crashdemons> \*n=1 to the word length or n=0 to length-1 (correction) | |  | |