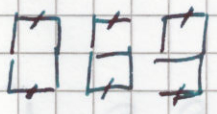
Read in 15 Strings : 0 - 9 input segments

10 delimiter

11 - 14 output segments

Compare all strings Length 5
" 6

all → 3 segments same
2,3 → + 1 segment same
3,5 → + 1 segment same



all → 2 segments same
0,6 → + 3 segments same

~~b~~	~~c~~	~~d~~	~~f~~	b	e	(2/5)
~~a~~	g	~~c~~	~~d~~	~~f~~	a	(2/5)
a, b	~~f~~	b	~~c~~	a	~~d~~	(3)
~~a, b~~						

c	d	f	b	e	
g	c	d	f	a	
f	b	c	a	d	
~~c~~ g	e	f	a	b	d
c	d	f	g	e	b
c	a	g	e	d	b

a = 4
b = 5
c = 6
d = 6
e = 4
f = 5
g = 3

⑧ a c e d g f b

c d f b e

g c d f a

f b c a d

⑦ d a b

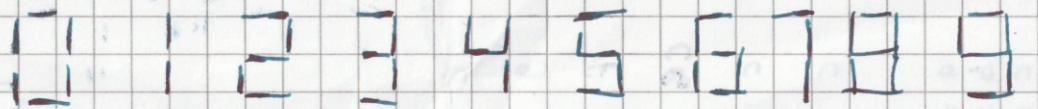
c e f a b d

c d f g e b

④ e a f b

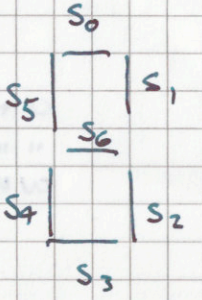
c a g e d b

① a b



len(2) = ①, gives S_1 and S_2 ↑ we know from counts!

len(3) = ⑦, difference with ① gives S_0



sort lengths at same time?

- Get count of freq. of letters in inputs → Solve S_0, S_2, S_4, S_5

$S_1 = 8$

[4, 6, 7, 7, 8, 8, 9]

- ~~a = 8~~
- b = 9 → S_2
  c = 7
  d = 8
- e = 6 → S_5
- f = 7
  g = 4 → S_4

S_0 → from counts (=8)
S_1 → from counts / ①
S_2 → from counts (=9)
S_3 → remaining letter!
S_4 → from counts (=4)
S_5 → from counts (=6)
S_6 → from counts / ④

0 => 0111111 => 63
1 => 110 => 6
2 => 1011011 => 91
3 => 1001111 => 79
4 => 1100110 => 102
5 => 1101101 => 109
6 => 1111101 => 125
7 => 0000111 => 7
8 => 1111111 => 127

S_6	S_5	S_4	S_3	S_2	S_1	S_0
64	32	16	8	4	2	1

9 => 1101111 = 111

- Letters into value of segments
- Sum values
- Convert to display value = output.