

```
# Write code to generate multiples of 7 until 1000 such that they are
stored in a list in this manner:
```

```
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]
```

Each element is a tuple and the first value of each tuple is the index and 2nd value is the multiple of 7

```
# Solution 1 Gautam
```

```
>>> l1= [i for i in range(0, 1000, 7)]
>>> for index, value in enumerate(l1):
    print(index, value)
```

```
0 0
1 7
2 14
3 21
4 28
5 35
.
.
```

```

.
140 980
141 987
142 994
>>> # [(0, 0), (1, 7), .... , (142, 994)] we want like this
>>> l1
[0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112,
119, 126, 133, 140, 147, 154, 161, 168, 175, 182, 189, 196, 203, 210,
217, 224, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294, 301, 308,
315, 322, 329, 336, 343, 350, 357, 364, 371, 378, 385, 392, 399, 406,
413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497, 504,
511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602,
609, 616, 623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693, 700,
707, 714, 721, 728, 735, 742, 749, 756, 763, 770, 777, 784, 791, 798,
805, 812, 819, 826, 833, 840, 847, 854, 861, 868, 875, 882, 889, 896,
903, 910, 917, 924, 931, 938, 945, 952, 959, 966, 973, 980, 987, 994]
>>> l_2 = [(index, value) for index, value in enumerate(l1)]
>>> l_2
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

```

```

# Solution 2 Seema Milind

```

```

>>> n_1=[i for i in range (1000) if i%7==0]
>>> n3=[(index,value) for index,value in enumerate(n_1)]
>>> n3
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]
>>> n3 == l_2
True

```

Solution #3 Shubham

```

>>> l= [(i, 7*i) for i in range(int(1000/7))]
>>> 1000/7
142.85714285714286
>>> int(1000/7)
142
>>> temp_1 = int(1000/7)
>>> l= [(i, 7*i) for i in range(temp_1)]
>>> for i in range(temp_1):
    print(i)

```

0

1

```

2
3
4
5
6
7
.
.
.
139
140
141
>>> # generate index then value ###Shubham's method
>>> for i in range(temp_1):
    print(i, 7 * i)

```

```

0 0
1 7
2 14
3 21
4 28
5 35
.
.
.
139 973
140 980
141 987
>>> l= [(i, 7*i) for i in range(temp_1 + 1)]
>>> l
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,

```

```

602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

```

Solution 4

```

>>> m = [ i for i in range(0,1000) if i%7==0]
>>> m
[0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112,
119, 126, 133, 140, 147, 154, 161, 168, 175, 182, 189, 196, 203, 210,
217, 224, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294, 301, 308,
315, 322, 329, 336, 343, 350, 357, 364, 371, 378, 385, 392, 399, 406,
413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497, 504,
511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602,
609, 616, 623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693, 700,
707, 714, 721, 728, 735, 742, 749, 756, 763, 770, 777, 784, 791, 798,
805, 812, 819, 826, 833, 840, 847, 854, 861, 868, 875, 882, 889, 896,
903, 910, 917, 924, 931, 938, 945, 952, 959, 966, 973, 980, 987, 994]
>>> n = [ (m.index(x) , x) for x in m]
>>> n
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,

```

```

763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

```

```

>>> # First generate value then index ###Meghnath
>>> L = [(i,int(i/7)) for i in range(1000) if i%7==0]
>>> L
[(0, 0), (7, 1), (14, 2), (21, 3), (28, 4), (35, 5), (42, 6), (49,
7), (56, 8), (63, 9), (70, 10), (77, 11), (84, 12), (91, 13), (98,
14), (105, 15), (112, 16), (119, 17), (126, 18), (133, 19), (140,
20), (147, 21), (154, 22), (161, 23), (168, 24), (175, 25), (182,
26), (189, 27), (196, 28), (203, 29), (210, 30), (217, 31), (224,
32), (231, 33), (238, 34), (245, 35), (252, 36), (259, 37), (266,
38), (273, 39), (280, 40), (287, 41), (294, 42), (301, 43), (308,
44), (315, 45), (322, 46), (329, 47), (336, 48), (343, 49), (350,
50), (357, 51), (364, 52), (371, 53), (378, 54), (385, 55), (392,
56), (399, 57), (406, 58), (413, 59), (420, 60), (427, 61), (434,
62), (441, 63), (448, 64), (455, 65), (462, 66), (469, 67), (476,
68), (483, 69), (490, 70), (497, 71), (504, 72), (511, 73), (518,
74), (525, 75), (532, 76), (539, 77), (546, 78), (553, 79), (560,
80), (567, 81), (574, 82), (581, 83), (588, 84), (595, 85), (602,
86), (609, 87), (616, 88), (623, 89), (630, 90), (637, 91), (644,
92), (651, 93), (658, 94), (665, 95), (672, 96), (679, 97), (686,
98), (693, 99), (700, 100), (707, 101), (714, 102), (721, 103), (728,
104), (735, 105), (742, 106), (749, 107), (756, 108), (763, 109),
(770, 110), (777, 111), (784, 112), (791, 113), (798, 114), (805,
115), (812, 116), (819, 117), (826, 118), (833, 119), (840, 120),
(847, 121), (854, 122), (861, 123), (868, 124), (875, 125), (882,
126), (889, 127), (896, 128), (903, 129), (910, 130), (917, 131),
(924, 132), (931, 133), (938, 134), (945, 135), (952, 136), (959,
137), (966, 138), (973, 139), (980, 140), (987, 141), (994, 142)]

```

Solution 5 Muzzamil

```

>>> L = [(i // 7, i) for i in range(1000) if i%7==0]
>>> L
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,

```

```
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]
```

```
>>> k = [(i , v) for i,v in enumerate(range(0, 1000, 7))]
```

```
>>> k
```

```
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
```

```

(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]
>>> m = [ i for i in range(0,1000) if i%7==0]
>>> m
[0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112,
119, 126, 133, 140, 147, 154, 161, 168, 175, 182, 189, 196, 203, 210,
217, 224, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294, 301, 308,
315, 322, 329, 336, 343, 350, 357, 364, 371, 378, 385, 392, 399, 406,
413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497, 504,
511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602,
609, 616, 623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693, 700,
707, 714, 721, 728, 735, 742, 749, 756, 763, 770, 777, 784, 791, 798,
805, 812, 819, 826, 833, 840, 847, 854, 861, 868, 875, 882, 889, 896,
903, 910, 917, 924, 931, 938, 945, 952, 959, 966, 973, 980, 987, 994]
>>> n = [ (m.index(x) , x) for x in m]
>>> n
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

```

#### One liners of previous solutions

```

>>> n = [ ([i for i in range(0,1000) if i%7==0].index(x) , x) for x
in [ i for i in range(0,1000) if i%7==0] ]

```



```

>>> n
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,
560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

>>> n_1=[i for i in range (1000) if i%7==0]
>>> n3=[(index,value) for index,value in enumerate(n_1)]
>>> n3=[(index,value) for index,value in enumerate([i for i in range
(1000) if i%7==0])]
>>> n3
[(0, 0), (1, 7), (2, 14), (3, 21), (4, 28), (5, 35), (6, 42), (7,
49), (8, 56), (9, 63), (10, 70), (11, 77), (12, 84), (13, 91), (14,
98), (15, 105), (16, 112), (17, 119), (18, 126), (19, 133), (20,
140), (21, 147), (22, 154), (23, 161), (24, 168), (25, 175), (26,
182), (27, 189), (28, 196), (29, 203), (30, 210), (31, 217), (32,
224), (33, 231), (34, 238), (35, 245), (36, 252), (37, 259), (38,
266), (39, 273), (40, 280), (41, 287), (42, 294), (43, 301), (44,
308), (45, 315), (46, 322), (47, 329), (48, 336), (49, 343), (50,
350), (51, 357), (52, 364), (53, 371), (54, 378), (55, 385), (56,
392), (57, 399), (58, 406), (59, 413), (60, 420), (61, 427), (62,
434), (63, 441), (64, 448), (65, 455), (66, 462), (67, 469), (68,
476), (69, 483), (70, 490), (71, 497), (72, 504), (73, 511), (74,
518), (75, 525), (76, 532), (77, 539), (78, 546), (79, 553), (80,

```

```

560), (81, 567), (82, 574), (83, 581), (84, 588), (85, 595), (86,
602), (87, 609), (88, 616), (89, 623), (90, 630), (91, 637), (92,
644), (93, 651), (94, 658), (95, 665), (96, 672), (97, 679), (98,
686), (99, 693), (100, 700), (101, 707), (102, 714), (103, 721),
(104, 728), (105, 735), (106, 742), (107, 749), (108, 756), (109,
763), (110, 770), (111, 777), (112, 784), (113, 791), (114, 798),
(115, 805), (116, 812), (117, 819), (118, 826), (119, 833), (120,
840), (121, 847), (122, 854), (123, 861), (124, 868), (125, 875),
(126, 882), (127, 889), (128, 896), (129, 903), (130, 910), (131,
917), (132, 924), (133, 931), (134, 938), (135, 945), (136, 952),
(137, 959), (138, 966), (139, 973), (140, 980), (141, 987), (142,
994)]

```

```

#####
#

```

Remove duplicates in a list [WIP]

```

#####
#

```

```

>>> l_1 = [1, 2, 3, 4]
>>> l_2 = [1, 2, 8, 7]
>>> l_3 = [3, 4, 8, 7]
>>> for i in l_1:
    if i in l_2: # Check presence
        print(i)

```

1

2

>>>Did not work

```

>>>for i in l_1:
    if i not in l_2:
        print(i)

```

3

4

```

>>> for i in l_1:
    if i not in l_2:
        print(i)
    elif i not in l_1: # Never executed at all as all values is
from l_1
        print(i)

```

```
3
4
>>> for i in l_1:
      if i not in l_2:
          print(i)
```

```
3
4
>>> for i in l_2:
      if i not in l_1:
          print(i)
```

```
8
7
>>>
```