ISO 6346 Check Digit Algorithm

Index	Omitted	Letter	Number
0	0	**0**	**0**
1	0	**1**	**1**
2	0	**2**	**2**
3	0	**3**	**3**
4	0	**4**	**4**
5	0	**5**	**5**
6	0	**6**	**6**
7	0	**7**	**7**
8	0	**8**	**8**
9	0	**9**	**9**
10	0	**A**	**10**
11	1	**B**	**12**
12	1	**C**	**13**
13	1	**D**	**14**
14	1	**E**	**15**
15	1	**F**	**16**
16	1	**G**	**17**
17	1	**H**	**18**
18	1	**I**	**19**
19	1	**J**	**20**
20	1	**K**	**21**
21	2	**L**	**23**
22	2	**M**	**24**
23	2	**N**	**25**
24	2	**O**	**26**
25	2	**P**	**27**
26	2	**Q**	**28**
27	2	**R**	**29**
28	2	**S**	**30**
29	2	**T**	**31**
30	2	**U**	**32**
31	3	**V**	**34**
32	3	**W**	**35**
33	3	**X**	**36**
34	3	**Y**	**37**
35	3	**Z**	**38**

Position	Weighting	Code	Lookup	Product
1	**1**	Z	38	38
2	**2**	E	15	30
3	**4**	P	27	108
4	**8**	U	32	256
5	**16**	0	0	0
6	**32**	0	0	0
7	**64**	3	3	192
8	**128**	7	7	896
9	**256**	2	2	512
10	**512**	5	5	2560
			Sum	**4592**
			Check	**5**

Remainder	Digit
0	**0**
1	**1**
2	**2**
3	**3**
4	**4**
5	**5**
6	**6**
7	**7**
8	**8**
9	**9**
10	**0**

Weighting is the base two exponent of the *Position* (beginning at zero.)

Lookup is is corresponding *Letter* to *Number* mapping for each *Code*.

Product is the *Weighting* multiplied by the *Lookup*.

Check is the remainder of the *Sum* of the *Products* divided by 11. If the *Check Digit* is 10, then it is mapped to 0 using a modulus operation.

Index is a range 0–35.

Omitted is the quotient of the *Index* minus one, all divided by ten.

Letter is a range 0–Z. It is derived by a lookup in the ASCII table. If the *Index* is less than 10, add 48 to resolve a number. Otherwise, add 55 to resolve a letter.

Number is mapped by adding *Omitted* to the *Index*, thus skipping multiples of 11 including 22 and 33. Note that the dashed lines are where these multiples would have been had they not been omitted.