

## Error Detection Using Check Digits

### UPC

The final digit of a Universal Product Code is a check digit computed as follows:

1. Add the digits (up to but not including the check digit) in the odd-numbered positions (first, third, fifth, etc.) together and multiply by three.
2. Add the digits (up to but not including the check digit) in the even-numbered positions (second, fourth, sixth, etc.) to the result.
3. Take the remainder of the result divided by 10 (modulo operation) and subtract this from 10 to derive the check digit.
4. If the last digit of the result in step 2 is 0, then the check digit is 0.

For example, the UPC-A barcode "036000241457" has last digit "7" as the check digit. If the other numbers are correct then the check digit calculation must produce 7.

1. We add the odd number digits:  $0+6+0+2+1+5 = 14$
2. Multiply the result by 3:  $14 \times 3 = 42$
3. We add the even number digits:  $3+0+0+4+4 = 11$
4. We add the two results together:  $42 + 11 = 53$
5. To calculate the check digit, take the remainder of  $(53 / 10)$ , which is also known as  $(53 \text{ modulo } 10)$ , and subtract from 10. Therefore, the check digit value is 7.

### ISBN 10

The final character of a ten digit International Standard Book Number is a check digit computed so that multiplying each digit by its position in the number (counting from the right) and taking the sum of these products modulo 11 is 0. The furthest digit to the right (which is multiplied by 1) is the check digit, chosen to make the sum correct. It may need to have the value 10, which is represented as the letter X. For example, take the ISBN 0-201-53082-1. The sum of products is  $0 \times 10 + 2 \times 9 + 0 \times 8 + 1 \times 7 + 5 \times 6 + 3 \times 5 + 0 \times 4 + 8 \times 3 + 2 \times 2 + 1 \times 1 = 99 \equiv 0 \text{ modulo } 11$ . So the ISBN is valid.

This can be validated very simply by adding all the products together then dividing by 11. If the final sum is a multiple of 11, then the ISBN is valid.

### ISBN 13

ISBN 13 (which began use in 2007) is equal to the EAN-13 code found underneath a book's barcode. Its check digit is generated the same way as the UPC, except the even digits are multiplied by 3 instead of the odd digits.

## LUHN Algorithm

The *Luhn algorithm* or *Luhn formula*, also known as the “modulus 10” or “mod 10” algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers. It was created by IBM scientist Hans Peter Luhn and described in U.S. Patent No. 2,950,048, filed on January 6, 1954, and granted on August 23, 1960.

The algorithm is in the public domain and is in wide use today. It is not intended to be a cryptographically secure hash function. It was designed to protect against accidental errors, not malicious attacks. Most credit cards and many government identification numbers use the algorithm as a simple method of distinguishing valid numbers from collections of random digits.

The formula verifies a number against its included check digit, which is usually appended to a partial account number to generate the full account number. This account number must pass the following test:

1. Counting from the check digit, which is the rightmost digit, and moving left, double the value of every second digit, modulo 9.
2. Sum the digits of the products together with the undoubled digits from the original number.
3. If the total ends in 0 (put another way, if the total modulo 10 is equal to 0), then the number is valid according to the Luhn formula; else it is not valid.

Assume an example of an account number "4992739871" that will have a check digit added, making it of the form 4992739871x:

Number	4	9	9	2	7	3	9	8	7	1	x
Double every other	4	18	9	4	7	6	9	16	7	2	x
Sum digits	$4 + (1 + 8) + 9 + (4) + 7 + (6) + 9 + (1 + 6) + 7 + (2) = 64 + x$										

To make the sum divisible by 10, the check digit must be set to 6, making the full account number 49927398716.

The account number 49927398716 can be validated as follows:

1. Double every second digit, from the rightmost:  $(1 \times 2) = 2$ ,  $(8 \times 2) = 16$ ,  $(3 \times 2) = 6$ ,  $(2 \times 2) = 4$ ,  $(9 \times 2) = 18$
2. Sum all the *individual* digits (digits in parentheses are the products from Step 1):  $6 + (2) + 7 + (1+6) + 9 + (6) + 7 + (4) + 9 + (1+8) + 4 = 70$
3. Take the sum modulo 10:  $70 \bmod 10 = 0$ ; the account number is probably valid.

### Wikipedia Sources

[http://en.wikipedia.org/wiki/Error\\_detection\\_and\\_correction](http://en.wikipedia.org/wiki/Error_detection_and_correction)  
<http://en.wikipedia.org/wiki/Checksum>  
[http://en.wikipedia.org/wiki/Check\\_digit](http://en.wikipedia.org/wiki/Check_digit)  
[http://en.wikipedia.org/wiki/Luhn\\_algorithm](http://en.wikipedia.org/wiki/Luhn_algorithm)