

## Book Number Nifty Assignment(s)

Book Number involves ISBN, the International Standard Book Number, which is a special code printed on the books of most countries on earth.

The ISBN code consists of 9 digits followed by a tenth check symbol. This check symbol serves to detect errors if some of the 9 digits are improper. The algorithm relating the digits to the check symbol begins by summing the first digit plus 2 times the second digit plus 3 times the third, and ultimately 9 times the ninth digit.

Then compute the remainder of this sum divided by 11.

If this remainder is 10 then the check symbol is "X"

otherwise the check symbol is the remainder.

For example, the Book Number 020508005 has check symbol 7, and the Book Number 123456789 has the check symbol 'X'.

This algorithm can be implemented in many ways as follows.

### 1. Book Number with Many Integers (9 of them)

Given 9 decimal integers (from 0 to 9 only)

with values either input or assigned such as:

```
d1 = 1; d2 = 2; ... d9 = 9;
```

compute and print out the check symbol.

### 2. Book Number as One Integer, with a series of arithmetic steps

Given a single integer (without leading zeros), such as :

```
int iISBN = 123456789;
```

break it apart into it's digits (left to right, then also right to left),

by a series of simple divide (/) multiply (\*) and modulo (%) operations, to compute and print out the check symbol. Do 3 ways and compare.

What if the book number has a leading zero such as:

```
int oISBN = 012345678;
```

### 3. Book Number as One Integer, done with a loop

Given a single integer (without leading zeros), such as :

```
int iISBN = 20508005;
```

break it apart into it's digits, (left to right, or right to left; choose one),

using a loop (While or For loop; choose one) to compute the check.

What if the book number has a leading zero?

### 4. Book Number as a real value (double)

Given a single real number, with the first digits (representing the language; English is 0) separated by a decimal point such as :

```
double rISBN = 1.23456789;
```

compute the check symbol again.

## 5. Book Number as a String

Given a string representing a book number, such as:

```
String sISBN = "123456789";
```

break it apart into its digits (can be done many ways), and compute and print out the check symbol.

Modify this if any number of dashes (or other non digits) are used, such as:

```
String sISBN = "1-234-56789"
```

and compute and print out the check symbol.

## 6. Book Number as an Array of Digits

Given an array representing a book number, such as:

```
int [] aISBN = {1,2,3,4,5,6,7,8,9};
```

or alternately:

```
int [] aISBN = {0,1,2,3,4,5,6,7,8,9};
```

compute and print out the check symbol.

## 7. Book Number Methods (static)

Encase each of the above computations as methods:

a void method which prints out the check symbol,

a string typed method which returns a check character.

and a boolean method which checks if a code and given check symbol match.

Do a method recursively.

## 8. Book Number Object

Define a BookNumber object consisting of the decimal digits and including the check symbol, and create a constructor for each of the different forms.

Create various kinds of methods including a boolean method which indicates whether a given BookNumber is of the proper book form.

## 9. Book Number Tree

Create a tree from the 9 ISBN digits by combining the digits from right to left, each time forming an inner node having the sum of the joined digits.

Continue this joining, to discover that the check sum is the sum of all internal digits. Notice that this process avoids any multiplication.

(You may need to draw a tree to see how and why this works).

## 10. Book Number as Stack, Queue, Linked List, etc

Create an algorithm to compute the check symbol by "pouring" digits from one stack into another, first reversing the order, then incrementing, and finally by summing the contents of a stack (similar to the two pass array).

Do this similarly with queues (one queue is sufficient).

## 11. Alternate Book Code Algorithms

Other ways of computing the check symbol are possible.

Show or prove that the following way works also.

Find the sum of 10 times the first digit, (at the left) and

9 times the second digit, 8 times the third digit,

etc, to 2 times the ninth digit.

The check sum is  $(11 - (\text{sum} \% 11)) \% 11$ .

Yet another way to compute the check symbol

from the array implementation is to loop through the array twice.

The first pass increments each value by the sum of the previous values,

(starting at the right) and the second pass simply sums all of these values.

For example:

```
int[] isbn = {0,2,0,5,0,8,0,0,5};
```

The first pass (from right to left) yields the array

```
aISBN = {20,20,18,18,13,13,5,5,5}
```

The second pass sums these to yield 117.

This way does not use any multiplication.

Why does this way work?

## 12. More on Book Numbers: on the internet

The 10 digit ISBN is to be replaced by a newer 13 digit version by year 2007;

Find it on the internet and compare these two kinds of book numbers

Find the books having the above book numbers.

Bar codes have similar algorithms; find some.

More on BookNumber can be found at:

[www.csun.edu/~jmotil/NiftyBookNumber](http://www.csun.edu/~jmotil/NiftyBookNumber)