Write an application that can convert a product ID number to a standard ISBN-10 number.

If we take the Da Vinci Code as an example, the product ID is 978140007917 and the ISBN is 1400079179. The first 3 digits of the product ID (978) are a prefix that can be removed. The remaining digits of the product ID (140007917) are the digits of the ISBN excluding the error control digit.

Refer to **Appendix 1** below for a description of how an ISBN number is constructed.
Your task is to develop an application that can accept a Product ID number and generate the ISBN-10 number.

You can code this up as a console application that takes in Product ID as input then outputs the ISBN.

Sample Test Cases

ProductID	**ISBN**
978155192370	155192370x
978140007917	1400079179
978037541457	0375414576
978037428158	0374281580

Appendix 1

EE4243 Data Communications

Secrets of the ISBN - An Error Detection Method

The International Standard Book Number (ISBN) uniquely identifies a published book with a ten digit number. For example, the book **Coding and Information Theory** by Richard Hamming, which was used as reference for this discussion, is uniquely coded as:

ISBN 0-13-139139-9

The familiar ISBN holds a secret.

In common with many similar numbers used in credit cards, price tags, inventory control and membership numbers, the ISBN is specially constructed to incorporate error detection. A study of the method used reveals many fundamental features of a robust error detection scheme (i.e. one which reliably detects errors).

Error control codes are often designed for known applications where certain types of errors are expected to occur. In this case, the most common errors expected would be those which humans would typically make when writing or typing a book order. These errors would normally be either "write a single digit incorrectly", or "switch two adjacent digits". Of course, the ability to detect errors also allows identification of invalid numbers which might be encountered (for example, if someone were to forge a credit card number). Forgery is not expected to be a real concern for book numbers, but error detection is very important.

The final digit in the ISBN provides the desired error detection capability. It is added once the book number is issued, and, once complete, the entire ten digit ISBN can be checked for errors whenever it is transcribed or transmitted.

ISBN 0-13-139139-9

The final digit is added for error control.

One Possible Approach - Checksum "Modulus 10"

Consider a simple scheme in which the first nine digits are simply added up and the sum (or rather the 'ones' digit in the sum) is used as the final ISBN digit. For example with ISBN 0-13-139139-_, the final digit would be '0':

```
ISBN          0    1    3    1    3    9    1    3    9    0
   sum        0 +  1 +  3 +  7 +  3 +  9 +  1 +  3 +  9 = 30
```

If any single digit is written incorrectly, this scheme will detect that error; however, if two digits are switched with each other, the sum will not be affected, and that error would go undetected. Although this simple 'checksum' does provide some error detection, it is not good enough for this application.

Modulus Arithmetic

Note that taking only the last digit in a decimal number is equivalent to dividing the number by 10 and using only the remainder. This result is also refered to the sum "modulus 10". This is generally written as:

30 = 0 mod 10

After division by 10, the remainder is 0.

How ISBN Actually Works

If a 'weighted sum' is constructed by multiplying each digit by a different constant, the problem of swapped digits going undetected can be eliminated. Furthermore, the final digit will be computed so that the weighted sum is an exact multiple of 11. In other words, the final digit is computed such that the sum "modulus 11" equals zero. In this case, 143 = 13 x 11, an exact multiple of eleven.

```
 ISBN          0    1    3    1    3    9    1    3    9    9
 multiplier  x10   x9   x8   x7   x6   x5   x4   x3   x2   x1
               --   --   --   --   --   --   --   --   --   --
    sum        0 +  9 + 24 +  7 + 18 + 45 +  4 +  9 + 18 +  9 = 143
```

143 = 0 mod 11

The ISBN is correct if a zero result is obtained.

Note that the remainder of "division by eleven" could be any number from 0 to 10. If it works out that the value of the final digit must be 10, the letter 'X' is used instead to complete the ISBN.

Why Modulus 11?

To explore the logic behind choosing 11 as a modulus, apply the above algorithm using 'modulus 5',

```
 ISBN          0    1    3    1    3    9    1    3    9    1
 multiplier  x10   x9   x8   x7   x6   x5   x4   x3   x2   x1
               --   --   --   --   --   --   --   --   --   --
    sum        0 +  9 + 24 +  7 + 18 + 45 +  4 +  9 + 18 +  1 = 90 = 0
mod 5
```

Two observations can be made:

1. The final digit could be either 1 or 6 and still give 0 mod 5.
2. The digit 9 (highlighted) could be anything and still give 0 mod 5.

It is clear that a digit greater than 10 must be used. As a final experiment, repeat the above algorithm using 'modulus 12',

```
 ISBN          0    1    3    1    3    9    1    3    9    X
 multiplier  x10   x9   x8   x7   x6   x5   x4   x3   x2   x1
               --   --   --   --   --   --   --   --   --   --
    sum        0 +  9 + 24 +  7 + 18 + 45 +  4 +  9 + 18 +  10 = 144 =
0 mod 12
```

This does not work perfectly either:
The final digit 9 could be 9 - 6 = 3 and the result would be 132 = 0 mod 12
The final digit 3 could be 3 + 4 = 7 and the result would be 156 = 0 mod 12
In either case, the error would go undetected.

These errors go undetected because 12 has factors (1,2,3,4,6) and any of these digit positions could contain a different digit which gives some other multiple of 12.

This problem can only be overcome if a prime integer is used (no factors). With no alternate method of obtaining a result which is an exact multiple of 11, the ISBN is guaranteed to detect all single-digit errors and all swapped-digit errors. Only the choice of a suitable prime modulus assures this result.

TWO DIGITS EXCHANGED

What if adjacent digits are accidentally swapped (i.e. 34 becomes 43) somewhere in the ISBN?

Let the two digits be A and B let them be at position (multiplier) N and N+1 respectively.

Their contribution to the correct sum is C0 = AN + B(N+1). If the two are swapped, their contribution becomes C1 = BN + A(N+1). In other words when these digits are exchanged, the ISBN sum simply changes by C1 - C0 = (A - B), regardless of where in the number these two adjacent digits appear. Since an ISBN error will go undetected only if the sum changes by a multiple of 11 ,and since the difference (A-B) can never equal 11, then adjacent swapped digits will always be detected.

In this example, the sum 143 changes to 149 when the digits 3 & 9 (highlighted) are swapped. The error is detected.

ISBN	0	1	3	1	**3**	**9**	1	3	9	9	
multiplier	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	
	--	--	--	--	--	--	--	--	--	--	
old sum	0 +	9 +	24 +	7 +	18 +	45 +	4 +	9 +	18 +	9	= 143
new sum	0 +	9 +	24 +	7 +	54 +	15 +	4 +	9 +	18 +	9	= 149

$$149 = 6 \bmod 11$$

The error is detected.

Further analysis shows that for two digits separated by a distance M the difference will be M(A-B), which can never be a multiple of eleven for M less than eleven. (Adjacent digits are a special case of M=1.) *This works because 11 is a prime number.* Consequently, the exchange of *any* two digits in the ISBN would be infallibly detected.

RULE FOR CHOOSING A MODULUS

It follows that the choice of 11 is a consequence of three rules:

1. The modulus must be greater than the number of digits in the ISBN.
2. The modulus must be greater than the range of digits in each position.
3. The modulus must be a prime integer.

WORST CASE PERFORMANCE

If an ISBN is totally garbled (or forged!), the use of modulus 11 implies that there is one chance in eleven that the (random) number will nonetheless pass the error detection test. Conversely, there are 10 chances out of 11 that the error *will* be detected. Consequently, while single-digit errors and swapped digits can be detected with 100% certainty, other types of errors should be detected with 10/11 = 91% reliability.

CONCLUSIONS

A systematic approach to error detection can lead to better codes. Both the simple checksum and the ISBN method involve adding only one additional digit, yet the ISBN method is more effective.

The need for a prime number as a divider is illustrated by the ISBN coding method. This observation is important in error control coding, and it is used in more sophisticated methods involving long binary messages. For example, the Cyclic Redundancy Check (CRC) is based on the mathematics of prime division.

EXAMPLE EXERCISES

Check the number ISBN 0-07-007013-X

ISBN	0	0	7	0	0	7	0	1	3	X

```
multiplier  x10   x9   x8   x7   x6   x5   x4   x3   x2   x1
            --   --   --   --   --   --   --   --   --   --
   sum       0 +  0 + 56 +  0 +  0 + 35 +  0 +  3 +  6 + 10 = 110
```

Recall that 'X' used when '10' is required in the final digit.

Since 110 = 0 mod 11, this ISBN is correct.

Complete the number ISBN 1-55512-010-_

```
ISBN          1    5    5    5    1    2    0    1    0    ?
multiplier  x10   x9   x8   x7   x6   x5   x4   x3   x2   x1
            --   --   --   --   --   --   --   --   --   --
   sum      10 + 45 + 40 + 35 +  6 + 10 +  0 +  3 +  0 +  ? = 149
```

Since 149 is not a multiple of 11, and the next closest multiple is 14x11= 154, the final digit should be 5.

Sum = 154 = 0 mod 11, for the valid ISBN 1-55512-010-5

22 JAN 98 - tervo@unb.ca
University of New Brunswick, Department of Electrical and Computer Engineering