



CMSHN1201 Programming Workshop Assignment 1

ISBN Check Digits

Coursework : Exam weighting

Weighting towards all courseworks (%)

Learning Outcomes Being Assessed

<ul style="list-style-type: none"> Modular programming techniques Software development techniques

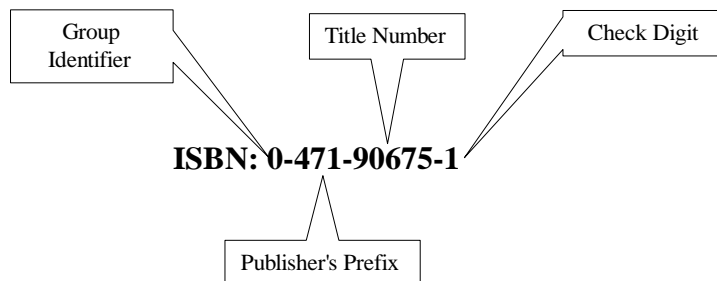
Set by	P. Vickers (718) <i>p.vickers@livjm.ac.uk</i>
Moderated by	A. Laws (603) <i>a.laws@livjm.ac.uk</i>
Hand-out date	5 October, 2000
Hand-in date	27 October, 2000
Expected hand-back date	8 November
Expected feedback date(if different from hand-back date)	
Estimated number of hours of student effort	10 Hours

Outline of Problem To write a C program that will:

- Allow the user to validate and calculate check digits for ISBN numbering.

Introduction

Most newly published books include an ISBN (International Standard Book Number) printed in the book and on the back cover. The number always contains ten digits divided into four parts:



The first three parts of the number are composed of digits in the range 0..9. The tenth digit is the check digit and must be in the range 0..9 or X (with X representing 10). The check digit is introduced to ensure that the previous nine digits have been correctly transcribed. The check digit is calculated in the following manner:

Multiply the first nine digits by 10, 9, 8, 7,...2 respectively, summing the results to give an integer sum, reduce *sum* modulo 11 to obtain an integer between 1 and 11 inclusive and subtract this from 11 to obtain the check digit.

For example, for the number 0-471-90675 given above, *a* is obtained by summing:

$$= (0 \times 10) + (4 \times 9) + (7 \times 8) + (1 \times 7) + (9 \times 6) + (0 \times 5) + (6 \times 4) + (7 \times 3) + (5 \times 2)$$

$$= 0 + 36 + 56 + 7 + 54 + 0 + 24 + 21 + 10$$

$$= 208$$

Reducing this modulo 11 (take the remainder from dividing by 11) gives integer *b* = 10 and 11 - 10 gives a check digit of 1.

Requirements Specification

Your task is to design and write a program (see Appendix A) that displays a menu with three options:

1. Calculate a check digit
2. Validate a ten-character ISBN
3. Quit

When the user selects option 1, the program should allow the user to type in a nine-digit ISBN. The program should then call a function `CheckDigit` that calculates the check digit for the given ISBN. The complete ten-character number should then be displayed on the screen with appropriate text, and formatted as 999999999-9. The user should be prompted to hit any key to return to the main menu.

Option 2 allows the user to enter a complete ISBN (without the hyphens). Using the function `CheckDigit` the program should display a message stating whether or not the number is valid.

The main menu for the program, then, will look like:

```
ISBN Program
-----

1 : Calculate check digit
2 : Validate a ten-character ISBN
3 : Quit

Select an action :
```

The function `CheckDigit`

The function `char CheckDigit (char [11])` is used to calculate a check digit for an ISBN. You pass the function a ten-character string holding the ISBN and it returns a character representing the check digit ('0'..'9', 'X'). Because we're reading the ISBN as a string, we need to convert the individual digits into integers for calculating the check digit. To help you on the way, consider the following code:

```
char x = '2' ;
printf ("%c : %d", x, x) ;
```

What is the output from this code? Notice that we can display a character as itself or as a number. What does this number mean? (see page 66 in the book). What does the following code output?

```
printf ("%c : %d", x, x - 48) ;
```

Editing and Compiling the Program

Unlike the programs you have seen to-date, this program uses 'separately compiled subprograms' (more of which in chapter 18 of the book). The function `CheckDigit` is not defined in the main program file (`menu.c`) but exists in a separate file `CheckDigit.c`. To help you get started, we have provided you with a working version of the program (called `isbn_pv.exe`) as well as a compiled version of `CheckDigit` (in `checkdigit.obj`). In other words, we coded the function and compiled it into a separate file. So, you can write the code for the main menu of the program, insert the appropriate call to the `CheckDigit` function and test your menu by linking to the external `.OBJ` file during compilation. This means that you can get the menu working properly before worrying about the `CheckDigit` function. It also means that having got the menu working, you can then write function.

So, you should tackle this coursework incrementally (in stages). The program you will eventually compile is called `isbn.exe`. The program is made up by linking two files:

- `menu.c`—The C source file that you will be editing

- **CheckDigit.obj**— The compiled file that contains a working version of the function *CheckDigit*

We link code from different files by using something called a **project**. On the module web page you will find a file called *isbn.ide* which is the **project file** for the program.

You can find the project file, *menu.c*, and *checkdigit.obj* files on the module's web site. You will notice that the OBJ file **does not** contain C source code.

Because this program (at least during the development stages) makes use of functions stored in separately compiled "object" files (.OBJ) compiling it requires a different set of actions to compile it. You can compile the program by doing the following:

1. Copy the project file (*isbn.ide*), the object file *checkdigit.obj*, and the main program file (*menu.c*) onto the 'M' drive.
2. Run the Borland C++ v5.02 compiler
3. Select Project|Open from the menu and use the Open Project File dialogue box to locate and open the project file *isbn.ide*.
4. On the screen you should see two windows. The left-hand window should contain the file *menu.c* and the right-hand window should show a hierarchical structure with *isbn.exe* at the top and *menu.c* and *checkdigit.obj* listed beneath it.
5. Make the necessary changes to *menu.c*
6. When you're ready to try compiling the program, right-mouse-click on *isbn.exe* in the Project window on the right. From the menu that appears on the screen select Build Node. The compiler will then try to compile the program. If all goes well, it should eventually give you a 'Success' message.
7. You can run the program (*isbn.exe*) by double-clicking on *isbn.exe* in the project window.

Writing the function

So, you've written the code for the main menu and tested it and now you're ready to tackle writing the function *CheckDigit*. How do you do it? Well, if you examine *menu.c* you will see at the bottom a section of code that has been enclosed by comment brackets. If you look carefully you will observe that this commented-out section contains a skeleton for the *CheckDigit* function. So, to write your own version of *CheckDigit*, here's what you do:

1. Remove the comment brackets from around the function definition.
2. Enter the necessary lines of C code to complete the function.
3. Remove *checkdigit.obj* from the project window (the one on the right) by right-mouse-clicking on *checkdigit* and selecting Delete node from the pop-up menu. This ensures that the version of the function that you have just written will be used instead of the version we provided.
4. Compile the project as before.

Deliverables

You should submit the following items by the deadline:

- A clearly labelled, virus-checked diskette with no sub-directories containing in the root directory the source file for your solution (*menu.C*)
- A clearly-labelled *listing* of your source program printed on separate sheets of paper suitably collated and stapled (fan-fold stationery must be separated and stapled) inserted into your workbook.

Note, any diskette found to be infected with a virus **WILL NOT BE ACCESSED** and hence your work cannot be assessed fully.

Remember, if you do not put your name and registration number on **all** items, then we will not know to whom we should award a mark. The first thing we do is to put the diskettes into a separate box from the listings. Hence, if your name is not on the diskette and we need to compile your program then we won't be able to do it and you lose marks.

Guidance

The program is interactive and involves lots of prompts and user data entry. A problem with data entered at the keyboard is that one must press the ENTER/RETURN key to submit the values. The

problem is that that causes a special end-of-line character also to be inserted into the data-read-buffer which, if not dealt with, will screw up any following input. The following example shows how use of an extra **char** variable solves the problem. Use this technique for accepting user input in the main menu:

```
int number ;
char dummy ;
...
...
printf ("Enter a number) : ") ;
scanf ("%d%c", &year, &dummy) ;
```

This puts the end-of-line marker into the char variable *dummy* hence removing it from the keyboard buffer.

All the files associated with the coursework can be downloaded from the module web site. There you will also find a fully working version of the program, called *isbn_pv.exe*. Running the fully-working version will give you a feel for what the specification really means.

If you have any problems, then make full use of the staff in the laboratory sessions and the tutorial. Above all, try to enjoy it. Programming is fun! It can also be mind-numbingly frustrating at times, but hey, you didn't come to university just to let your brain atrophy.

Resources Required

Access to C compiler

Assessment
Marks will be awarded for:

Assessment Criteria	Weight
<ul style="list-style-type: none"> • Compilation without errors • Producing the correct results with output exactly as specified. • Producing a correct design. • Structure of code matching outline design. • Use of meaningful identifiers and layout of code. • Use of correct data types 	100%

This is an individual piece of work

Appendix A Program Outline

```
#include <stdio.h>
#include <conio.h>          /* Needed for getch ()          */
main ()
{
  char isbn [11] ;
  char CheckDigit (char [11]) ; /* While working on the main bit, simply use */
                                /* checkdigit.obj in your project file      */

  int choice ;
  char dummy ;

  do
  {
    /* Display menu here, accept user choice, and act on it. */
  }
  while (    ) ;

  /* Put your version of CheckDigit here. */
  /*
  char CheckDigit (char isbn [11])
  {
  */
```




CMSHN102 Programming Workshop Coursework Cover Sheet

Complete and attach this form to your coursework submission.
THIS IS NOT A RECEIPT.

Name

Reg. No.

This work is for the attention of

Checklist:

The grade I realistically think I would get for this piece of work is

This work is my own; I have properly attributed any contributions by others where they occur and I understand the regulations surrounding collusion, plagiarism and cheating

I have included the following items (tick boxes that apply):

Virus-free diskette containing C source code:

Listing on collated stationery of your source program

Signed

Date