Project #2 – Book Cipher

Course	INFO-1156 Object-Oriented Programming in C++
Professor	Garth Santor, Lianne Wong, and Janice Manning
Assigned	March 19th 2018
Due	April 6th 2018 @ 11:59pm
Weight	5%
Student Name	

Project Description

Create two C++ console applications; one that encodes a text message using a book cipher, and another that decodes the message coded by the previous program.

Program Interfaces

The encoder should have the following command-line interface:

```
bcencode.exe bookfile messagefile codedfile
```

The decoder should have the following command-line interface:

```
bcdecode.exe bookfile codedfile messagefile
```

Where:

- `bookfile` is the name of the ASCII text file containing the book used to encode/decode the message.
- `messagefile` is the name of the ASCII text file containing the message to be encoded
- `codedfile` is the name of the ASCII text file containing the offset numbers

Encoding rules

Encode each character of the source file by replacing it with two numbers, the first representing the line in the book (zero-based), and the second the offset to that character in that line (also zero-based). Any method of choosing which offset to use is acceptable so long as the same offset isn't used repeatedly.
At the end of running bcencoder, a codedfile is created with offsets.
At the end of running bcdecoder, a messagefile is created with the message.

The encoded file must be just a file of the exact offsets of the message file.
bookfile: "ABCDEFGHIJKLM\nNOPQRSTUVWXYZ"
messagefile: "ACER"
codedfile: 0x00000000 0x00000000 0x00000000 0x00000002 0x00000000 0x00000004 0x00000001 0x00000004

Encoding background

To read more about encoding messages using the book cipher see:

https://en.wikipedia.org/wiki/Book_cipher
In this program, you are encoding letters, not the words. The encoded file will store the number in binary.

bctest.exe

'bctest.exe' is a program that can perform some tests on your encoder and decoder programs. The test program should be placed into the same folder as your executable files. If your programs are properly called 'bcencode.exe' and 'bcdecode.exe' the test program will automatically execute your programs at appropriate times during the test.

Grading Criteria

Functional Requirements		
Encoder works	40%	40%
Decoder works	40%	40%
Both programs report errors if any of the input files cannot be opened (indicating which file was the problem).	10%	10%
The encoder program reports and quits if the message cannot be encoded by the book file (i.e. A character in the message to be encoded does not exist in the book file), and reports which character cannot be encoded.	10%	10%
Non-functional requirements		
Executables program are **not** named 'bcencode.exe' and 'bcdecode.exe'	-10% / each	
Coded message uses the same index repeatedly for the same letter.	-20%	
Penalties from *C & C++ Grading Guide* ***v1.1.0*** A guide to code/comment style can be found at C++ Style	Various	
Late submission • One to five days late • More than five days late	 -10%/day -100%	
Total	**100%**	**100%**

Submission

1. Submit **entire Visual Studio project directory** to Fanshawe Online
    a. Delete ***all*** debug and release directories.[i]
    b. Submit in a .ZIP, .7z archive file.

[i] Alternatively, you can 'clean' your project for submission by downloading 'vsclean' a Visual Studio Solution Cleaner from www.gats.ca .