

## Assignment 3: Cobol Re-engineering (20%)

### ISBN Validation

The International Standard Book Number ( ISBN ) is a code that uniquely specifies a book. It is based on the 9-digit Standard Book Number (SBN) created by Gordon Foster for the booksellers W.H. Smith in 1966. The number is either 13 or 10 (before 1 January 2007) digits long. For example here are the two ISBNs for the book “*Tuscany: Simple Meals & Fabulous Feasts from Italy*”.

ISBN-10 : 1784881198

ISBN-13 : 9781784881191

The 10-digit ISBN was developed by the International Organization for Standardization and released in 1970. Since January 2007, ISBNs have contained 13-digits, a format to enable conformance with the EAN global numbering system (a barcode standard), and to allow for an increase in the available number of ISBNs. Ironically most books published today have both 10 and 13 digit ISBNs.

### DESCRIPTION

In both the 10-and 13-digit ISBN, the last digit of the ISBN is known as the check digit. Its value is calculated from the other digits in the ISBN, and its purpose is to check the validity of the ISBN. If a mistake is made in copying the ISBN, the resulting ISBN will be invalid, which indicates an error.

For the 10-digit ISBN, the rightmost digit is a checksum digit, which can be uniquely determined from the other nine digits from the condition using the equation:

$$x = 10d_1 + 9d_2 + 8d_3 + 7d_4 + 6d_5 + 5d_6 + 4d_7 + 3d_8 + 2d_9$$

where  $d_i$  denotes the  $i$ th digit of the ISBN from the left. The checksum digit  $d_{10}$  can have any value from 0 to 10: the ISBN convention is to use the value X to denote 10. The resulting value is used to calculate  $x \bmod 11$ , which is the remainder on division of  $x$  by 11. The resulting modulus is subtracted from 11, giving the check digit. If the check digit is 10, it is replaced with “X.” For example, the calculation for the 10-digit ISBN whose first nine digits are 0-201-31452 is

$$\begin{aligned} &= 10 \times 0 + 9 \times 2 + 8 \times 0 + 7 \times 1 + 6 \times 3 + 5 \times 1 + 4 \times 4 + 3 \times 5 + 2 \times 2 \\ &= 0 + 18 + 0 + 7 + 18 + 5 + 16 + 15 + 4 \\ &= 83 \end{aligned}$$

The value of  $83 \bmod 11$  is 6, and so  $11 - 6 = 5$ . Therefore, the check digit is 5, and the complete sequence is 0-201-31452-5

Here is an example for validating ISBN 1784881198:

1	7	8	4	8	8	1	1	9
×10	×9	×8	×7	×6	×5	×4	×3	×2
10	63	64	28	48	40	4	3	18

The results add up to:

$$10 + 63 + 64 + 28 + 48 + 40 + 4 + 3 + 18 = 278$$

Modulo 11 the result:

$$278 \bmod 11 = 3$$

Subtract from 11:

$$11 - 3 = 8$$

The results equal the check digit which is 8. The ISBN is valid.

## TASKS

1. Create a modern Cobol program to perform ISBN validation of a series of 10-digit ISBNs stored in a user-inputted file.
2. Include three “subprograms” in the form of paragraphs:
  - 2.1. **readISBN** - Prompts the user for the name of an ASCII file containing the list of ISBN numbers. Reads the values of the ISBN numbers and processes them. If the file does not exist, the program should produce an error message and re-prompt for the filename.
  - 2.2. **isValid** - Checks the validity of the ISBN, i.e. whether or not it contains characters it shouldn't. Responses should include indication of whether a number contains erroneous characters.
  - 2.3. **checkSUM** - Extracts the individual digits, and calculates the checksum digit.
3. Produce an output for each ISBN number in the file, identifying whether it is valid or not.

## TESTING

The program will be tested on a file a series of ISBN numbers, of the form:

1856266532	correct and valid
0864500572	correct and valid with leading zero
0201314525	correct and valid with leading zero
159486781X	correct and valid with trailing uppercase X
159486781x	correct and valid with trailing lowercase X
0743287290	correct and valid with leading and training zero
081185213X	correct and valid with leading zero, trailing X
1B56266532	incorrect, contains a non-digit
159486781Z	incorrect, contains a non-digit/X in check digit
1856266537	correct, but not valid (invalid check digit)

## HINTS

- The 10-digit ISBN is not always a pure number. The check digit can often be an X - design accordingly.
- The ISBN does not need to be checked for length, because of the way COBOL specifies input. Only 10-character values will exist in the test files.

## COMPILING

Please do not include a Makefile, and make sure your program compiles in the following manner:

```
> cobc -free -x -Wall isbn.cob
```

# WRITE-UP INFORMATION

## REFLECTION REPORT

Discuss your re-designed program in 1 (one) page (or more if you like) **reflection report** (single-spaced), explaining decisions you made in the re-engineering process. Consider this document a synopsis of your experience with your Cobol re-engineering process. You could do this by answering the following questions:

- Given your knowledge of programming, was Cobol easy to learn?
- What structures made Cobol challenging to program in?
- What did you (i) like, (ii) dislike about Cobol?

## DELIVERABLES

The submission should consist of the following items:

- The reflection report (PDF).
- The code (well documented and styled appropriately of course):  
**isbn.cob**
- Both the code and the reflection report should be submitted as a ZIP, TAR, or GZIP file.

## STYLING & COMMENTS

Style consists of mnemonic variable names, indentation, and the use of whitespaces and paragraphing. The purpose of good style is to make the meaning of your program clear to someone who has never seen it before, cannot run it, and cannot talk with you. Documentation consists of in-code documentation. Examples of qualities to look for include:

- Are variable names well chosen?
- Are comments relevant rather than simple repetitions of the code?
- Do comments point out key sections of code, indicate special cases, or make assertions?
- Are the indents 3 or 4 spaces? Do not use tabs or 2 space indenting (please check “convert tabs to spaces” in your editor)
- Is whitespacing used to separate parts of the program to provide clarity?

## SKILLS

- This is an exercise in writing a Cobol program from scratch.