# ISBN-13 Programming Challenge

International Standard Book Numbers-13 (IGCSE Computer Science textbook, 2nd Ed., p. 60)

1. Complete the function `ValidateISBN13(ISBN)` so that the pseudocode below works.
2. **Trace** your pseudocode function so to ensure it works as expected; you may use examples from our textbook and ISBNs from your textbooks and library materials.
3. You may also want to **test** some of your pseudocode ideas using Pseudocode.Pro (Note that Pseudocode Pro's INT function does not convert strings to integers, though!)

```
////////////////////////////////////////////////
// IGCSE Pseudocode to input and validate ISBN-13 //
// REMEMBER that indices start at 1 in IGCSE...  //
// pseudocode, while they start with 0 in Python//
////////////////////////////////////////////////

// Input: ISBN-13 number as a string
FUNCTION ValidateISBN13(ISBN : STRING) RETURNS BOOLEAN
    DECLARE Sum, Digit, CheckDigit, i, Weight, Remainder, LastDigit : INTEGER
    // Your code goes here
ENDFUNCTION

// Finds the position of a character in a string
// equivalent to InputString.find(Character) in Python
FUNCTION FindCharacter(InputString :STRING, Character :STRING) RETURNS INTEGER
    FOR i ← 1 TO LENGTH(InputString)
        IF SUBSTRING(InputString, i, 1) = Character
          THEN
            RETURN i
        ENDIF
    NEXT i
    RETURN -1 // Character not found
ENDFUNCTION

// Replaces all occurrences of OriginalChar to NewChar
// Creates NewString based on OriginalString with the replacements done
// Similar to str.replace(old, new) in Python
FUNCTION ReplaceCharacter(OriginalString :STRING, OriginalChar :STRING,
NewChar :STRING) RETURNS STRING
    DECLARE NewString : STRING
    FOR i ← 1 TO LENGTH(OriginalString)
        IF SUBSTRING(OriginalString, i, 1) = OriginalChar
          THEN
            NewString ← NewString + NewChar
          ELSE
            NewString ← NewString + SUBSTRING(OriginalString, i, 1)
        ENDIF
    NEXT i
    RETURN NewString
ENDFUNCTION
```

```
// "Main" ------------------------------------------------------------------
DECLARE ISBN : STRING
DECLARE Number, CheckDigit : INTEGER

REPEAT
    INPUT ISBN
    IF FindCharacter(ISBN, "-") <> -1
      THEN
        ISBN ← ReplaceCharacter(ISBN, "-", "") // remove dashes
    ENDIF
UNTIL LENGTH(ISBN) = 13 AND ISNUMERIC(ISBN)
// assume ISNUMERIC(string) behaves like ISBN.isnumeric() in Python,
// i.e. returns true if the string can be converted to an int/has only numbers
// assume STRING_TO_INT(argument) converts its argument into an integer
Number ← STRING_TO_INT( SUBSTRING(ISBN, 1, LENGTH(ISBN)-1) )
CheckDigit ← STRING_TO_INT( SUBSTRING(ISBN, LENGTH(ISBN)-1, LENGTH(ISBN)) )

IF ValidateISBN13(ISBN)
  THEN
    OUTPUT "Valid ISBN-13!"
  ELSE
    OUTPUT "INVALID ISBN-13"
ENDIF
```

4. Now write the Python code corresponding to the pseudocode above, plus your answer. You may use string slicing instead of the substring function, and *str.isnumeric()* for type validation.

5. Instead of using Python's string functions ***find*** and ***replace*** functions, write your own ***find_character*** and ***replace_character*** functions. You may use the corresponding provided pseudocode functions as inspiration.

6. Modify your Python code so that the user input is validated with the ISBN-13 check digit; the user will get an error message and will have to re-input the string until it passes the check digit validation.

7. Test your code with a few ISBN-13 numbers from your school textbooks/novels and intentional mistakes introduced into their ISBNs to make sure your solution recognises both valid and invalid ISBN-13s correctly.