Practical 5
Calculator (Part 2)

Step 1

Write the **`Convert`** subroutine that converts a string of characters into a 15-bit unsigned integer with error handling.

Input : **A0.L** points to a string.

Outputs : **Z** returns false (0) if an error occurs. That is to say, if the string:

- is empty.
- contains at least one character that is not a digit.
- represents an integer higher than 32,767.

Otherwise **Z** returns true (1) (no conversion error).
If **Z** returns false, then **D0.L** is not modified.
If **Z** return true, then **D0.L** returns the integer value of the string.

Tips:

`Convert` is similar to **`Atoui`** but with error handling. Therefore, you should check if the string is valid and call **`Atoui`** if it is.

Step 2

Write the **`Print`** subroutine that displays a string of characters on the video output window.

Inputs : **A0.L** points to a string to display.
D1.B holds the column number where the string will be displayed.
D2.B holds the line number where the string will be displayed.

Tips:

- The video output window of the debugger can be shown by pressing **`[F4]`**.
- To use the video output window, you have to slightly modify the vector initialization as follows:

```
                    org     $0

vector_000          dc.l    $ffb500
vector_001          dc.l    Main
```

Do not try to understand this modification for the time being.

- The subroutine **`PrintChar`** is at your disposal. It displays a single character on the video output window. To use it, you must copy the `"PrintChar.bin"` file in the same folder as your source file and include the following line:

```
PrintChar           incbin  "PrintChar.bin"
```

PrintChar has the following inputs:

Inputs : **D0.B** holds the ASCII code of the character to display.
D1.B holds the column number where the character will be displayed.
D2.B holds the line number where the character will be displayed.

- Use **PrintChar** to display successively each character of the string on the video output window.

Use the following structure in order to run and test your subroutine:

```
                    ; ==============================
                    ; Vector Initialization
                    ; ==============================

                    org     $0

vector_000          dc.l    $ffb500
vector_001          dc.l    Main

                    ; ==============================
                    ; Main Program
                    ; ==============================

                    org     $500

Main                lea     sTest,a0
                    move.b  #24,d1
                    move.b  #20,d2
                    jsr     Print

                    illegal

                    ; ==============================
                    ; Subroutines
                    ; ==============================

Print               ; ...
                    ; ...
                    ; ...

PrintChar           incbin  "PrintChar.bin"

                    ; ==============================
                    ; Data
                    ; ==============================

sTest               dc.b    "Hello World",0
```

Step 3

Write the **`NextOp`** subroutine that returns the memory location of either the first operator in a string or the null character if no operators are found. The string can contain any types of characters (letters, punctuation, digits, operators, etc.).

Input : **A0.L** points to a string.

Output : **A0.L** returns the address of the first operator in the given string or the address of the null character if no operators are found.

Ex.: Before:

A0 ↓									
'1'	'0'	'4'	'+'	'9'	'*'	'2'	'-'	'3'	0

After:

			A0 ↓						
'1'	'0'	'4'	'+'	'9'	'*'	'2'	'-'	'3'	0