Pratical 7
Calculator (Part 4)

For this practical, the **`GetInput`** subroutine is put at your disposal. It allows you to get a string of characters keyed in by a user. **`GetInput`** has the following inputs:

Inputs : **A0.L** points to a 60-byte buffer where the user string will be stored.
D1.B holds the column number where the user string will be displayed.
D2.B holds the line number where the user string will be displayed.
D3.L holds the time delay index before the first repetition.
D4.L holds the time delay index after the first repetition.

The buffer should be reserved by the `DS.B` assembly directive.

- Type the source code below and save it under the name "`GetInputTest.asm`".
- Copy the "`GetInput.bin`" file into the same folder.

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

                    org     $0

vector_000          dc.l    $ffb500
vector_001          dc.l    Main

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

                    org     $500

Main                movea.l #sBuffer,a0
                    clr.b   d1
                    clr.b   d2
                    move.l  #60000,d3
                    move.l  #8000,d4
                    jsr     GetInput

                    illegal

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

GetInput            incbin  "GetInput.bin"

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

sBuffer             ds.b    60
```

- Run this code and display the video output window. Enter a string of characters and press the **[Enter]** key. Have a look at the contents of memory location `buffer`. Repeat the process until you grasp how the **GetInput** subroutine works. Be careful, you are not asked to execute **GetInput** step by step; you just have to understand how to use it.

Note:
The **D3** and **D4** parameters should be adjusted according to the performance of your computer. If the repetition is too fast, you should increase these values.

Step 1

Write the **Itoa** subroutine that converts a 16-bit signed integer into a string of characters.

Inputs : **A0.L** points to a buffer where the string will be stored after conversion.
D0.W holds the 16-bit signed value to convert.

For instance, if **D0.W** = `-10825`, the following string should be placed at the address held in **A0** :

A0
↓

'-'	'1'	'0'	'8'	'2'	'5'	0

Tips:

- If the number is positive, just call **Uitoa**.
- If the number is negative, put the `'-'` character in the string and call **Uitoa** with the additive inverse of the number.

Step 2

Write the **Main** program of the calculator that complies with the following example:

```
Enter an expression:
2*5-5*2+18/2
Result:
14
```

Step 3

Modify your program so that the operator precedence is taken into account. Use the method of your choice.

```
Enter an expression:
2*5-5*2+18/2
Result:
9
```