CS 218
Homework, Asst. #6

Purpose:	Become familiar with data conversion, addressing modes, and assembly language macro's.
Due:	Wednesday (6/14)
Points:	75

Assignment:

As we know from television and movies, an alien invasion imminent. Based on a number of tabloid reports, the aliens have three fingers (see eyewitness sketch on right). As such, it is assumed they will use a base-3, or ternary[1], numbering system.

Write an assembly language program using macros to convert a ternary string into an integer, add the values, and convert the resulting integer to a ternary string and output the result. Specifically, write the following simple assembly language macros:

- Macro, **ternary2int**, that will convert an ASCII string (STR_SIZE characters, byte-size, signed, right justified, blank filled, NULL terminated) representing the signed ternary value into a signed double-word sized integer. *Note*, in order to convert a series of strings, this macro will be called repeatedly by the main.

- Macro, **int2ternary**, that will convert a signed double-word sized integer into an ASCII string representing the ternary value. The macro must store the result into an ASCII string (10 characters, byte-size, signed, right justified, blank filled, NULL terminated).

Use the provided main, which will invoke the macros. You may assume valid/correct data. As such, no error checking is required. For example, given the following initial test data set:

```
tlet0     db        "     +12012", NULL
ans0      dd        0
```

The string would be converted to integer, doubled, and converted to string for display. The result, as displayed to the screen, would as following:

```
-----------------------------------------
Answer:
    +101101
```

The other data sets are added (instead of doubling).

The ASCII strings are STR_SIZE (provided constant) characters long including the NULL. All data must be treated as *signed* integers. As such, the IMUL and IDIV instructions should be used (not the DIV or MUL). The provided program template includes a macro that prints the ASCII lists to the screen. No changes to the parameter lists are allowed. You may declare additional variables as needed. Do **not** change the data types (double-words or bytes) as defined. *Note*, since the program displays the results to the screen, typing `./ast06` will execute the program (assuming *ast06* is the program name). You can refer to a web based on-line based converter to check results.

---

1 Ternary is a base-3 numeral system. For more information, see: http://en.wikipedia.org/wiki/Ternary_numeral_system

Debugging:
The code for a macro will not be displayed in the source window. In order to see the macro code, display the machine code window (**View → Machine Code Window**). In the window, the machine code for the instructions are displayed. The step and next instructions will execute the entire macro. In order to execute the macro instructions, the **stepi** and **nexti** commands must be used.

Submission:
When complete, submit:

- A copy of the source file via the class web page.
  Assignments received after the start time of class will not be accepted.

Example Output:
The results, as displayed to the screen, would be as follows:

```
ed@ed-vm% ./ast6
-----------------------------------------------------
CS 218 - Assignment #6

-----------------------------------------
Answer:
    +101101

-----------------------------------------
Answer:
   +1100122

-----------------------------------------
Answer:
   +2210101

 [...output truncated due to space...]

ed@ed-vm%
```

Assignment #6 - Data

```
; -----
;  Assignment #6 Provided Data

tlet0     db        "     +12012", NULL
ans0      dd        0

tlet1     db        "     +12002", NULL
          db        "    +101212", NULL
          db        "   +1020201", NULL
          db        "      -1201", NULL
          db        "    -102022", NULL
len1      dd        5
ans1      dd        0

[...data listing truncated due to space...]
```

Note, the ".bss" section is for uninitialized data. The "resd" is for reserve double words and "resb" is for reserve bytes. See provided main for the rest of the data declarations.