```
; Count the occurrences of each letter (A to Z)
; in an ASCII string terminated by a NUL character.
; Lower case and upper case should be counted
; together, and a count also kept of all
; non-alphabetic characters (not counting the
; terminal NUL).

; The string starts at x4000.

; The resulting histogram (which will NOT be
; initialized in advance) should be stored starting
; at x3100, with the non-alphabetic count at x3100,
; and the count for each letter in x3101 (A) through
; x311A (Z).

; R0 holds a pointer to the histogram (x3100)
; R1 holds a pointer to the current position in the string
;    and holds the loop count during histogram initialization
; R2 holds the current character being counted
;    and is also used to point to the histogram entry
; R3 holds the additive inverse of ASCII '@' (0xFFC0)
; R4 holds the difference between ASCII '@' and 'Z' (xFFE6)
; R5 holds the difference between ASCII '@' and ''' (xFFE0)
; R6 is used as a temporary register

00110000 00000000     ; starting address is x3000

1110 000 011111111    ; LEA R0,xFF      (point R0 to the start of the histogram)

; fill the histogram with zeroes
0101 110 110 1 00000  ; AND R6,R6,#0   (put a zero into R6)
0010 001 000100000    ; LD R1,x20      (initialize loop count to 27)
0001 010 000 1 00000  ; ADD R2,R0,#0   (copy start of histogram into R2)
; loop to fill histogram starts here
0111 110 010 000000   ; STR R6,R2,#0   (write a zero into histogram)
0001 010 010 1 00001  ; ADD R2,R2,#1   (point to next histogram entry)
0001 001 001 1 11111  ; ADD R1,R1,#-1  (decrement loop count)
0000 001 111111100    ; BRp x1FC       (continue until loop count reaches zero)

; initialize R1, R3, R4, and R5 from memory
0010 011 000011011    ; LD R3,x1B      (set R3 to additive inverse of ASCII '@')
0010 100 000011011    ; LD R4,x1B      (R4 holds difference between ASCII '@' and 'Z')
0010 101 000011011    ; LD R5,x1B      (R5 holds difference between ASCII '@' and ''')
0010 001 000011011    ; LD R1,x1B      (point R1 to start of string)

; the counting loop starts here
0110 010 001 000000   ; LDR R2,R1,#0   (read the next character from the string)
0000 010 000010100    ; BRz x14        (found the end of the string)

0001 010 010 0 00 011 ; ADD R2,R2,R3   (subtract '@' from the character)
0000 001 000000100    ; BRp x4         (branch if > '@', i.e., >= 'A')

0110 110 000 000000   ; LDR R6,R0,#0   (load the non-alpha count)
0001 110 110 1 00001  ; ADD R6,R6,#1   (add one to it)
0111 110 000 000000   ; STR R6,R0,#0   (store the new non-alpha count)
0000 111 000001100    ; BRnzp xC       (branch to end of conditional structure)

0001 110 010 0 00 100 ; ADD R6,R2,R4   (compare with 'Z')
0000 001 000000101    ; BRp x5         (branch if > 'Z')

; note that we no longer need the current character
; so we can reuse R2 for the pointer to the correct
; histogram entry for incrementing
0001 010 010 0 00 000 ; ADD R2,R2,R0   (point to correct histogram entry)
0110 110 010 000000   ; LDR R6,R2,#0   (load the count)
0001 110 110 1 00001  ; ADD R6,R6,#1   (add one to it)
0111 110 010 000000   ; STR R6,R2,#0   (store the new count)
0000 111 000000101    ; BRnzp x5       (branch to end of conditional structure)

; subtracting as below yields the original character minus '''
0001 010 010 0 00 101 ; ADD R2,R2,R5   (subtract ''' - '@' from the character)
0000 110 111110011    ; BRnz x1F3      (if <= ''', i.e., < 'a', increment non-alpha)

0001 110 010 0 00 100 ; ADD R6,R2,R4   (compare with 'z')
0000 110 111110111    ; BRnz x1F7      (if <= 'z', go increment alpha count)
0000 111 111110000    ; BR x1F0        (otherwise, go increment non-alpha)

0001 001 001 1 00001  ; ADD R1,R1,#1   (point to next character in string)
0000 111 111101010    ; BRnzp x1EA     (go to start of counting loop)

1111 0000 00100101    ; TRAP x25       (done)

; the data needed by the program
00000000 00011011     ; 27 loop iterations
11111111 11000000     ; the additive inverse of ASCII '@'
11111111 11100110     ; the difference between ASCII '@' and 'Z'
11111111 11100000     ; the difference between ASCII '@' and '''
01000000 00000000     ; string starts at x4000
```