2.2 Function to Print 16-bit Numbers in Hex

Code a function to print the number from R0 in upper case hexadecimal. For this you could code rotates and mask operations to extract each of the 4 nibbles. It might also use a decision to tell whether the nibble is 0-9 or A-F. To obtain the character to print in the first case, just add the nibble with the ASCII code for 0. In the second case, add the nibble to the constant (ASCII code for A MINUS Ten). Calculate this constant when you write the program. (A production assembler will let you code \texttt{CONSTA10} \texttt{.FILL 'A'-#10}.)

You might alternatively use table lookup to translate a nibble into Hex.

Each Hex number printout should begin with \texttt{x}. The printout should NOT end with a newline.

The Hex printing function must not “mess up” any of the registers other than R0 because this function will be called within functions that are quite unrelated. Since it won’t be recursive, it can save register values in nearby memory words. R7 must be saved that way also because the LC-3 TRAPs you will call for printing will use R7.

If the DEBUG flag is set, the program should test your Hex printing function. A successful test is indicated by the printout:

\textbf{Testing Hex printer:xC3B0}

(You will get 0 for the project if you fake this or any other test!)

2.5 Main input and output

First the program should set the DEBUG flag. The Hex printing test should be done next if the DEBUG flag is 1.

Remember to write \texttt{BLKW ...} code to reserve memory for the stack, and \texttt{to make R6 point to the word after the bottommost stack word soon after the program begins to run.}

Debugging or not, the program should next print:

\textbf{Enter Pattern:}

and, on the same line, accept a pattern string of length up to 30 and store it as an LC-3 string (one character per 16-bit word terminated by 0.)

You should code an LC-3 function that inputs a string of length up to 30, with echoing, into memory beginning at the address given in R0. This function can be called once to get the Pattern and again to get the Subject.

If the user types more than 30 pattern characters, print “\texttt{Input too long.} and call the QUIT trap.

Your program should make the LC-3 “echo” the characters as the user types them because this is what all computer users expect when they type on the keyboard. See the PP textbook for how to do this.

A newline (ASCII value 10 or 0xA) OR a carriage return (ASCII value 13 or 0xD) will indicate that the user pressed the ENTER key. When either of these values are returned by the GETC trap, it should (like other characters) be echoed with OUT. Then, 0 (zero) should be stored after characters that had been before. Since the maximum string length is 30, you must provide 31 words to accommodate them plus the 0 terminator.

Then, do the analogous things to input the subject string, length up to 30:

\textbf{Enter Subject:}