

EE 371 LAB 5 F05
BINARY DATA DISPLAY

Artificial intelligence is no match for natural stupidity

Schedule: Oct 5,6: Do the lab

Name _____

Partner: _____

Demo _____

Extra(1) _____ (2) _____

Code _____

Memo _____

1. Our customer for last weeks assignment was very happy with the software you wrote. In fact, she has now given us another job to do
2. Specifications: The customer is trying to teach some students the difference between hexadecimal and binary (probably future Microsoft programmers!). She would like a program that does the following:
 - a. Prints out the message:
Enter a two digit hexadecimal number -> \$
 - b. Allows the user to type two hexadecimal digits on the keyboard. The user may type any character on the keyboard but the program is to accept ONLY the valid hexadecimal characters 0 - 9, A - F or a - f. If a valid character is typed, it should be echoed (printed) to the terminal. If an invalid character is typed, it should not be echoed, just ignored. When two valid characters are entered, the program should:
 - c. Print out the message:
The binary equivalent is %
with the correct binary equivalent.
 - d. Example:
Enter a two-digit hexadecimal number -> \$45
The binary equivalent is %01000101
3. Of course, as is the case with all *Two Dot Engineering* software, the *ECE Department Assembly Language Practice* document will be followed (see <http://www.coe.montana.edu/ee/courses/ee/ee371/ee371hpg.htm#standards>).
4. Helping Hand #1 You should use the serial I/O routines and putstr routine you used last week.
5. Helping Hand #2 Subroutines are required to be called from the main program. DO NOT put subroutines in the main program. One subroutine should capture the keystrokes and when two valid hex digits are entered, return the BINARY number. Another subroutine should take the binary number and print out the 1's and 0's. As you finish the detailed design, you may find other subroutines are necessary or useful.
6. Helping Hand #3 Converting ASCII-hex characters to hex digits requires the algorithm shown below:

```
      ; Convert a valid ASCII hex digit to hex
      ; IF the character is lower than $3A
      ; THEN subtract $30
      ; ELSE
      ; Force to upper case
      ; Subtract $37
      ; ENDIF
```

To verify how this works, check out the ASCII codes for the hex digits \$0-\$F.
7. Helping Hand #4 When you need to print out a byte in binary, you must check each bit starting from the most significant bit. The Rotate/Shift instructions will help you do this in conjunction with some of the conditional branching instructions. There are a variety of ways to do this.
8. Demonstration You are to demonstrate the program to the customer's representative in the lab. Make sure you can show it working for a variety of inputs.
9. Extra credit #1 Extra credit may be granted if your program allows the user to continue entering two digit hex numbers until the Escape key is entered at any time. Demonstrate to the customer representative how you are doing this.
10. Extra credit #2 Additional extra credit points may be granted if your input routine accepts the backspace character (BS=\$08) to allow the user to correct mistakes when entering data. When the BS character is received, the program should erase the character by printing the character sequence BS SP BS (\$08 \$20 \$08). The user should be allowed only as many backspaces as there are characters entered, i.e. don't backup over the \$ in the prompt. The user's input is to be terminated by typing the Enter key but once two valid digits are typed, all subsequent characters are ignored until the <Enter> (or a backspace) key is typed.
11. Grading: Hand in your list file for the subroutine written to capture keystrokes and return the binary number as specified in 5 - one list per group. Demo 10 points (**-2 points for each demo not working 100% correctly**), program .LST file 10 points [you will be graded on your use of labels, equates, orgs, comments and other assembler language features - only one listing per group need be submitted], first extra credit 5 points, second extra credit 5 points.
12. Memo Report: Each student is to write a memo report on the putstr program sent to you by email. The report is to describe how the subroutine works and is to comment on the adequacy of the comments from the original programmer. Please EMAIL the report to me by Wednesday October 12.