CPE 325: Embedded Systems Laboratory
Laboratory Assignment #1

Assignment [50 pts]

1. **[25 pts]** You are working as a government cryptographer and are tasked with writing a C program that implements a Caesar cipher to encrypt a string. The encryption scheme should take any string of characters and shift the ASCII value of alphabetical characters right by 3. Numbers, spaces, and special characters should be left alone. For example, if the given string is "**Hello all, welcome to CPE325 Fall 2024!**", the output should be like the following:

   Khoor doo, zhofrph wr FSH325 Idoo 2024!

   Note: You can hard code the given string as follows, you may be asked to change it during demonstration:

   char msg[]="Hello all, welcome to CPE325 Fall 2024!

   To print the string in a single line using printf, the following statement

   can be used: printf("%s\n", msg);

   Hint: The ascii code equivalent for 'a' is 0x61

   More about the cipher: https://en.wikipedia.org/wiki/Caesar_cipher

1. **[25 pts]** You have been given a program **Lab01_P2.c** that computes the prime factorization of a hard coded integer value. This program does not work in its current state: it has syntax errors (this will cause the project build to fail) and logical errors (once built & run, the program output is incorrect or program behaves atypically). You are free to make any changes necessary to correct the program, provided the correct output will still be given if the input value is changed. An example of correct program output is shown below:

   Input value: 126
   Output factors: 2 3 3 7

   Please also measure the number of clock cycles taken by the function call get_prime_factors() for at least two different input values, and include these measurements in your report. How does the input value affect the number of cycles taken?

Hints to get you started on debugging:

- You can easily check if your program output is correct by multiplying the output factors: these should equal the input value.
- Syntax errors for a program will be shown in the "Problems" window after the project is built. After the compiler finds its first syntax error, it may incorrectly parse the rest of the source file; for this reason, you should correct the first error in the list and then rebuild the project to see if there are any errors remaining.
- To debug the logical error, you can try the following:
  - Choose different input values, and compute their prime factorizations by hand. Compare your factorization with the output from your program, and form a hypothesis about the nature of the problem *(e.g. Are one or multiple output values not what they should be? Are there too many values? Too few?)*
  - Divide and conquer - break the problem down into smaller pieces *(e.g. If the printed output is wrong, is it because the program is printing it incorrectly, or because it was calculated incorrectly? Can you tell whether the returned values of num_factors, and the array values in prime_factors[], are correct?)*
  - Single-step through the program using your debugger and follow the values of variables of interest. Compare what you see in the debugger with the behavior you expect given the program comments.

Topics for Theory

1. Discuss the following tools/features from Code Composer Studio in your report
   a. Memory window
   b. Console window
   c. Variable window
   d. Breakpoints
2. What commands in Code Composer Studio can you use to run through your program?

Deliverables

1. Lab report which includes:
   a. Brief theory discussion
   b. A neat **flowchart** for each of the programs
   c. **Output screenshots** for both of the programs
   d. Source code (.c files) in appendix, included in lab report

Notes:

1. You must create an organized directory, subdirectory, workspace, and project for each demo code and each solution.
2. During demonstration, you should be able to inspect variables, set watchpoints, set and monitor breakpoints, monitor registers and memory, and show the output.
3. While comparing the results for part 2, make sure you test different input sets.
4. The report (PDF) should be a single submission with the source code pasted at the end of it.