

CMSC 216 Quiz 1 Worksheet

The first quiz for the course will be on Wed, Sep 18. The following list provides additional information about the quiz:

- The quiz will be a written quiz (no computer).
- The quiz will be in lab session.
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>
- Make sure you know your section number and your TA's name.

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours. It is recommended that you try these exercises on paper first (without using the computer). Qu

Exercises

1. What is the first name of your lab TA and what is your section number?
2. Be familiar with the information provided in the C style guidelines available at:
<http://www.cs.umd.edu/class/fall2013/cmsc216/resources/coding-style.html>
3. Name at least one difference between a #include and an import statement in Java.
4. Name and briefly explain the compilation stages associated with a C program.
5. What is the difference between static storage and automatic storage?
6. Write a Unix command that will copy all C files present in the directory /tmp to your home directory, assuming your current directory can be any directory.
7. Name two flags we use with gcc for compilation.
8. Write the Unix command that will allow us to compile the program myProg.c, and then leave the executable in a file named myProg.x.
9. Write a complete C program that reads two integer values and prints the powers of two of values in the specified range. You can assume the first value is less than or equal to the second. For example, if the user enters 3 and 4 we expect to see 8 16.
10. Write a C function that determines whether a positive sequence of integer values provided by the user represent an increasing sequence. For example, 3, 6, 10 represents an increasing sequence. The function will return true if the sequence is increasing and false otherwise. You can assume a negative value will mark the end of the sequence. You may not use arrays for this problem and your function must work for any number of values (not just 3).
11. Write a recursive C function that computes the n^{th} number in the Fibonacci sequence.
12. Expand the previous C function so you can print the number of times the function was called.