CMSC 216 Quiz 1 Worksheet

The first quiz for the course will be on Tue, Jun 5. 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.
- You must use a pencil.
- Quiz instructions can be found at [http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html](http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html)
- **Regarding Piazza** - Feel free to post questions in Piazza regarding the worksheet and possible solutions to problems.

**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 TAs or instructors during office hours. It is recommended that you try these exercises on paper first (without using a computer).

### Exercises

1. Make sure you have read the information available at:
   
   [http://www.cs.umd.edu/class/resources/academicIntegrity.html](http://www.cs.umd.edu/class/resources/academicIntegrity.html)

2. When is a project considered late in this class?

3. Write a Unix command that will list of the files present in the current directory that end with the .c extension.

4. Write a Unix command that will copy the file ex.txt present in the ~/tmp directory to:
   
   a. The current directory.
   b. Your home directory.
   c. To the parent’s directory of the current directory.
   d. To the root directory (you can assume you can copy to the root directory).

5. Write a Unix command that will copy the folder `commands_info` present in the `lecture_examples` folder of the ~/216public directory to your ~/216 directory of your home directory. The command should work when executed from any directory (e.g., you cannot assume your current directory is ~/216).

6. What is the size (in bytes) of a char type?

7. Suppose you write a C program and it has an infinite loop; how do you stop the program?

8. For this quiz, you will need to provide examples of academic integrity violations. The following is the list you need to know:
   
   a. Hardcoding of results in a project assignment. Hardcoding refers to attempting to make a program appear as if it works correctly (e.g., printing expected results for a test).
   b. Using any code available on the internet/web or any other source.
   c. Hiring any online service to complete an assignment for you.
   d. Sharing your code or your student tests with any student.
   e. Using online forums (other than Piazza) in order to ask for help regarding our assignments.

9. Does the following code compile? Briefly explain.

   ```c
   #include <stdio.h>
   
   void pl(int x) { printf("%d", x); }
   
   void pl(float x) { printf("%d", x); }
   
   int main() {
     pl(2);
     return 0;
   }
   ```
10. Define a function called `print_powers` (int `print_powers(int limit)`). For this problem:

- The function will read a character (either `f` or `i`). If the user enters `f`, the function will print the powers of numbers from 1 up to the `limit` value (specified in the parameter) in increments of .5. If the user enters `i`, the function will print the powers of values from 1 up to the `limit` value in increments of two.
- Use `scanf` to read the character. The function will display the following message in order to read the character: "Enter f (float) or i (integer):"
- You can assume users will enter correct data (either `f` or `i`).
- Notice the output format is important. See the example we have provided below. If the user enters `i`, data is displayed as integer values; otherwise as float values.
- You can assume the `limit` parameter will be greater than 1. Notice that the output might not include the limit value (e.g., user enters `i`, but the limit value is 6).

**The function will return how many powers were printed.**

The following driver and associated output illustrates the functionality expected from the function you need to write. Keep in mind this is just an example (your function must work for different sets of values and not just the ones presented in the example). In the example, underlined text is input the user provides and % is the Unix prompt. Notice we are running the program twice and we are not using the value returned by the function. For the first program execution, the function returns 9; for the second the function returns 3.

**Driver**

```c
int main() {
    print_powers(5);
    return 0;
}
```

**Output**

```
% a.out
Enter f (float) or i (integer): f
1.000000, 1.000000
1.500000, 2.250000
2.000000, 4.000000
2.500000, 6.250000
3.000000, 9.000000
3.500000, 12.250000
4.000000, 16.000000
4.500000, 20.250000
5.000000, 25.000000
% a.out
Enter f (float) or i (integer): i
1, 1
3, 9
5, 25
%```

11. Define a function called `compute` (prototype below) that computes either the sum or product of integers provided by the user. For this problem:

- The function reads integers values using `scanf` and the message "Enter value: " to read each value.
- The function will stop reading values when the user provides -1.
- If the `sum_flag` parameter is true, the function will compute and return the sum of the values; otherwise it will compute and return the product.
- If the `print_flag` parameter is true, the function will print the computed valued before returning the value. Use the message "Result: " followed by the computed valued.
- You can assume users will provide valid data (integers) and the computation will not cause an overflow.
- Below we provided an example of using the function. Underlined text represents input and % the Unix prompt.

**Driver**

```c
int main() {
    compute(0, 1);
    return 0;
}
```

**Output**

```
% a.out
Enter value: 4
Enter value: 5
Enter value: -1
Result: 20
%```

```c
int compute(int sum_flag, int print_flag)
```