

Homework 6: File I/O

CS16 - Winter 2021

Due:	Thursday, February 18, 2021 (11:59 PM PST)
Points:	80
Name:	-----
Homework buddy:	-----

- You may collaborate on this homework with **at most** one person, an optional “homework buddy.”
- **Submission instructions:** All questions are to be written (either by hand or typed) *in the provided spaces* and turned in as a single PDF on Gradescope. If you submit handwritten solutions write legibly. We reserve the right to give 0 points to answers we cannot read. When you submit your answer on Gradescope, **be sure to select which portions of your answer correspond to which problem** and clearly mark on the page itself which problem you are answering. We reserve the right to give 0 points to submissions that fail to do this.

1. (10 points) When testing for end of file, the lecture (and the readings) mention two methods. What are they and when is each best used?

2. (10 points) Consider the following code snippet from a program where everything is set up correctly:

```
cout << "Enter a line of input: ";  
char next;  
do {  
    cin.get(next);  
    cout << next;  
} while (!(isdigit(next)) && (next != '\n'));  
cout << "END!\n";
```

If the program dialog is as follows:

Enter a line of input: I'll see you 2nite at 9PM!

then what will be the next line of output? **And explain why.**

3. (10 points) I have a text file called “t.txt” that contains two entries: “University of California” on one line, and “Computer Science Rules!” on the next line.

Show the output produced when the following code (entire program not shown so assume all the necessary set ups are done correctly) is executed **and explain why that is**. You are encouraged to also try to compile this to verify your results.

```
ifstream tin;
char c;

tin.open("t.txt");

tin.get(c);
while (!tin.eof()) {
    if ((c != 'e') && (c != 'C')) {
        cout << c;
    }
    tin.get(c);
}
```

4. (10 points) Complete the code below (there are several missing lines) such that the program reads a text file called “MyInputs.txt”, which only contains double-type numbers separated by whitespaces. Of course, you don’t know ahead of time how many numbers are in the file. You **must** use all the variables declared below, and you may add more variables as needed. Your program should print the average of these numbers, as indicated below.

For example, if the text file contains this single line:

4.2 3.3 9.1 3.1 0 0 7.5 5.4 9.9 10

Then the program should print out:

The average is: 5.25

The code is as follows:

```
#include <iostream>

using namespace std;

int main ()
{
    ifstream inf;
    double num, sum(0), average;
    int count = 0;

    // ... (missing code) ...

    cout << "The average is: " << av << endl;

    return 0;
}
```

5. (40 points) Write a **function definition** for a function called `FindMedian()` that will read a file that just contains integer numbers that are separated by whitespaces and then finds the *median* value. The median of a set of numbers is the number that has the same number of data elements greater than the number as there are less than the number.

Requirements and Hints:

- You can assume that the input data text file will not have more than 100 numbers in it (but it could have fewer).
- You **have** to write the resulting median (just the integer and newline) in a separate output data file called `median_output.dat`. Don't print anything to standard output.
- The function does not have to check if the input or output files exist prior to reading/writing operations.
- The function definition should only have two arguments: the input file-name variable and the `ifstream` variable (see example function call below).
- Hint 1: It is relatively easy to find the median in a set of sorted integers. And, yes, it's OK to define additional functions!
- Hint 2: It matters if the number of integers is odd or even when finding the median. Think about how to address this.

The main function could look like this:

```
int main() {  
    ifstream ifs;  
    string fname = "inputs.txt"; // could have other names too...  
    FindMedian(fname, ifs);  
    return 0;  
}
```

Submission instructions:

- You will submit this question (just this question) on Gradescope under “HW 06 - Q5” as a C++ file called `median.cpp` (cannot be named otherwise). This should only contain any and all function definitions you have, **except** for `main` (we'll supply that). We will test your submission with an autograder. This has the same due date as this homework.