Final Review
CMSC 242
Spring 2017

Name: _____

1. One of the differences between C and C++ is that C++ supports objects while C only supports structs. List at least two differences between structs and objects and tell how a C programmer can simulate the missing features without using C++.

2. Matching: For each of the following system calls in the left column, write the letter corresponding to the description of that call in the right column.

fork	_____	a. Frees resources associated with a file descriptor.
exec	_____	b. Makes a copy of an open file and assigns it to a new file descriptor.
popen	_____	c. Creates a new process which is almost identical to the process that makes the system call.
select	_____	d. Copies a block of bytes to a file, stream, or pipe. Differs from "f" in that it doesn't take flags.
fopen	_____	e. Forks and execs a new process, mapping either the standard input or standard output to a pipe.
close	_____	f. Copies a block of bytes to a network stream. Differs from "d" in that it accepts a set of network-specific flags.
dup2	_____	g. Loads an executable file from disk into memory, replacing the memory space of the current process.
send	_____	h. Creates a buffered file stream.
write	_____	j. Blocks until activity occurs on one of several file descriptors, a signal is received, or a timeout occurs.
connect	_____	k. Establishes communication with a remote server.

(Note that there is no option "i")

3. How big an array of characters do I need to store the string “Yankee Doodle” if I plan to perform string operations (such as strlen) on it?

4. Write code for computing the “size” of a directory by opening the directory, looping through every entry in it, and summing together the size of each file. Don't forget to clean up after yourself.

5. What does the “atexit” system call do?

6. Explain the difference between the HOME and PATH environment variables by carefully describing what each one stores.

7. What does the SIGINT signal do and how can I generate it from the keyboard?

8. Write code for opening a file, copying twenty character values from an array named “ticket” to the end of the file, and the closing the file. You can assume the twenty characters are in positions 0 through 19 of the array.

9. Write code for opening a network connection to a server on system 192.168.50.99 using TCP port 3702, reading thirty characters from the stream, printing those bytes to standard output, and the closing the connection. Do not use any of the csapp.h functions – use only legitimate Unix system calls.

10. The function below takes one integer variable named “sock” which represents a properly connected network socket (obtained from “accept”). Write code that declares and properly initializes an fd_set variable which I could use as the “read set” in select. Give code that adds both “sock” and the standard input to the set.

```
void monitor_connection(int sock) {



        /* Select call here*/
}
```

11. What is one major difference between a thread and a process?

12. What does the “pthread_join” function do?

13. In detail, describe what happens when you try to access a page on the world wide web. In your explanation, describe what the browser sends to the server and also what the server sends to the client. **Give a complete example illustrating your answer.**

14. The function below uses pipes and “fork” to calculate the sum of two numbers in a child process (this might be done so that if the “summation” process crashes, it doesn't take down the parent process with it). Some of the code is missing (the part that involves the pipes). Add the missing code.

Hint: You will need to create TWO pipes (one for reading and one for writing), then use appropriate system calls to read from and write to them.

```
int add_nums()
{
        int pid;
        int x,  y;
        int result;

        //Set up the pipes




        pid = fork();
        if (pid == 0) {
                //Read two numbers from one pipe, add them, and write the result to the other pipe.




        }
        else {
                scanf(“%d %d”, &x, &y);
                //Write x and y to the first pipe, then read the result from the second pipe.




                printf(“The sum is: %d\n”, result);
        }
        //Don't forget to clean up after yourself at the appropriate place(s) in the code.


}
```

15. Suppose I have the following array of characters which represents a large text document:

char doc[40960];

Show how I could use “popen” and appropriate I/O calls to print the entire contents of the array to the screen one page at a time using the program “/usr/bin/more”.