CS221 C and Systems Programming – Study Questions

Instructor: Paul Haskell

1. What does the acronym "CPU" stand for?

_____

2. Fill in the following table. Assume all values are 16 bits long

	Binary	Octal	Hex
1			
-1			
10			
-10			
256			
16384			
-32768			

3. What are the 4 "memory regions" in a running program, and what is stored in each?

_____

4. What is the value of w?

```
short w = 16384 << 1;
```

_____

5. What is the `printf()` command to print a pointer variable?

_____

6. What is done by the C Preprocessor during compilation?

7. What is wrong with the following code?

```
struct Dog_t {
    char name[64];
    double age;
}
```

_____

8. Please write a typedef for the above `struct`.

_____

9. Write a Makefile that builds **prog.exe** from **prog.c**

_____

10. What is meant by "big-endian" and "little-endian"?

_____

11. Write the code to declare a pointer to `a1`

```
int main() {
    int a1 = 1, b2 = 2;

}
```

_____

12. Write code that generates a segmentation fault.

_____

13. What is `triple`?

```
double tripleMe(int inp) { return 3.0 * inp; }
double (*triple)(int) = tripleMe;
```

_____

14. Please write a simple C++ class.

---

15. Describe at least two multithreading "work flows".

---

16. What socket operations are done by a socket "server"? By a "client"?

---

17. Write a signal handler

---

18. Please write the structs needed for a basic singly-linked list.

19. Write a method that determines the endianness of the current computer and that prints either "big" or "little".

```
void whichEndian() {
```

20. Write a function that takes two pointers to integers and swaps their values.

```
void swap(int *a, int *b) {
```

21. Implement a function that takes a pointer to a dynamically allocated array and its size, then doubles the array size while preserving its contents. The function should return a pointer to the newly allocated array and should "clean up" the old array.

```
int* resizeArray(int *arr, int size) {
```

22. Write a function that removes all occurrences of the integer 'value' from an array and returns the new number of elements in the array. The array should be modified in place using pointers.

```
int removeElement(int *arr, int size, int value) {
```

23. Use `fscanf()` and `fopen()` to open a file "**Input.txt**", read each word, and print out the length of each word. The program should end cleanly at the end of the input file.

24. Write the RECURSIVE code for `reverseHelper()`

```
void reverseString(char* input) {
    char* end = input;
    while (end[0] != 0) { end++; }
    reverseHelper(input, end);
}
```

25. Write `myStrdup()`, your own implementation of the `strdup()` method.

26. How could you write a multithreaded program that gets stuck in a "deadlock" when each of two threads is stuck waiting for the other.

27. Write a method `getline()` that mimics the built-in `fgets(char*)` method, reading the keyboard input until it finds a NEWLINE and returning the resulting string.

28. Explain the difference between a pointer variable and a regular variable in C

29. Please use `malloc()` to allocate memory for an array of 20 `double` values

30. Describe how bitwise operations can be used to check whether a number is odd or even.

31. Write a short explanation of what happens when you dereference a NULL pointer in C.