CS214 Fall 2017 Midterm Exam

Name: _____________________________ NetID: _______________________________

Section (Circle 1): 1 2 3 4 5 6 7

Instructions:

- Do NOT begin until you are told to do so
- Do NOT remove the staple
- DO put your name on every sheet
- DO write legibly
- DO write concisely. Use the fewest number of words necessary to answer the question. No additional pages will be provided.

Just C things (Pick 10 out of 13 to answer. Circle the problem numbers you want graded):

1. What happens in each of the stages of compilation: Preprocessing, compilation, assembly, and linking?

2. The new version of Linux has a new system call called `groot`. How do you find what `groot` does and how to use it? Assume you do not have access to the internet.

3. What Is a system call? When do you use system calls?

4. How are strings represented in C? Create a string called `foo` with the value "cs214". This string must be mutable.

CS214 Fall 2017 Midterm Exam

Name: ______________________________ NetID: ________________________________

5. What is the difference between a struct, an enum, and a union? Which of these constructs is best suited to represent the colors of the rainbow? Instantiate such a construct In C called `colors`. The colors of the rainbow are red, orange, yellow, green, blue, indigo, violet.

6.

a) The following code attempts to reimplement `strcpy`. What is wrong with it? Show an example of how it goes wrong. You may assume initial addresses for dst and src to be 0x5000 and 0x7000 respectively.

```
void strcpy (char* dst, const char* src) {
   while (*src) {
         dst = src;
         dst++;
         src++;
   }
}
```

b) Edit the code above to fix the error. You may do this inline by crossing out and updating the code.

7. What is a segmentation fault? Name 2 different causes of segmentation faults.

CS214 Fall 2017 Midterm Exam

Name: ______________________________ NetID: ________________________________

8. Given this code:

```
unknown * thingy = (unknown*)malloc(4 * sizeof(unknown));

int mystery = 0;

mystery = (thingy + 1) – thingy;
```

What value does `mystery` hold?

9. Write a function pointer named "derp" for the following function:

```
int * oddFunction(int* values, struct stuff* storage, char delimiter)

{…}
```

10. Why might the following code segfault? Add some code to make sure it returns -1 rather than segfaults.

```
int aValue = 12;


int* ptr = (int*)malloc(4 * sizeof(int));


*ptr = aValue;
```

11. What are the differences between `strlen` and `sizeof` a string in C? Why? Show an example.

CS214 Fall 2017 Midterm Exam

Name: ____________________________ NetID: ______________________________

12. The code below is supposed to increment each value in an int array of length N by 1 and save the new value in a new array. What is printed out instead? Why? Fix the code so that the right thing happens. (Hint: the numbers in `someArray` are indeed incremented by 1 and stored somewhere)

```
while (i < N) {
        incrementArray[i] = someArray[i]++;
        printf("%d %d\n", incrementArray[i], someArray[i]);
        i++;
}
```

13. You wish to write a function that encrypts text as numerical values. You know that in C, memory is an amorphous entity. You wish to take every 4 characters in a string, and output the integer equivalent of those 4 bytes. E.g. the string "jack" is encoded as a single integer 1784767339. You may output via printf. You may assume that strlen(str) % 4 == 0. Do NOT make assumptions about the length of a string. Hint: This solution requires fewer than 10 lines of code.

j	a	c	K
01101010	01100001	01100011	01101011
01101010011000010110001101101011			
1784767339			

```
void convert (char* str) {
```

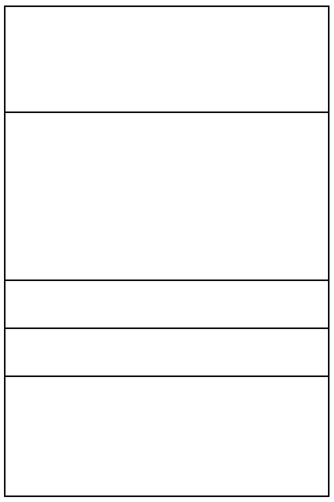
CS214 Fall 2017 Midterm Exam

Name: ____________________________ NetID: _______________________________

Memory Management (Answer all questions)

1. Fill in the following memory map with the correct labels. Then describe the function, in one sentence, of each part of memory. Possible labels: heap, stack, text, data, bss.

Oxfffffff



0x00000

2. What is `malloc()`? Why do C programs tend to have `malloc()` statements? What does `malloc()` return?

3. What is wrong with the following function?

```
int* sum (int a, int b) {
   int c = a + b;
   return &c;
}
```

Name: ______________________________ NetID: ________________________________

4.

a) Imagine a 32-bit system's implementation of `malloc` as an implicit list (size + free boundary tags). Given a 4096-byte block of memory to manage, and 100 successful `malloc` operations within that block (and no free operations), calculate the metadata overhead (e.g. amount of memory for metadata/total amount of memory). Assume size and the free tag are both stored as shorts.

b) On a 64-bit system, imagine that blocks are implemented in explicit lists (pointer + size + free boundary tags). What Is the metadata overhead now? Assume the same 4095 initial block and 100 malloc operations. Assume size and free are stored as shorts.

5. What is the benefit of an explicit free list in `malloc` implementations vs. implicit lists via size? What is a drawback of an explicit free list?

6. Presume your `malloc` implementation never checked for adjacent free blocks (coalesce) when freeing malloced memory. Given enough memory allocations and frees, eventually the code:

```
char* anArray = (char*)malloc(2 * sizeof(char));
```

would fail, no matter how much memory was being used. Why?

CS214 Fall 2017 Midterm Exam

Name: ____________________________ NetID: ______________________________

7. A buddy system memory allocator can coalesce adjacent free blocks quickly and reduces metadata overhead. What ways might a buddy system allocator waste more memory/than a first fit allocator with block splitting would? Max 4 sentences.

8. What is a memory leak? How does it occur? How does one fix it?

Project Redux (Answer all questions)

1. One of the issues faced by groups v/as to handle commas inside a movie title properly. How did you parse the string to ensure that movie titles with commas were parsed correctly?

2. Some groups used dynamically allocated arrays to store each movie record. Some groups used statically allocated arrays to store each movie record. Assume `Record` is a typedef struct representing all the fields of a movie record. Also assume `sizeof`(`Record`) returns 100.

```
Record arr[5000];

Record* arr2[5000];
```

   a) How much memory is allocated for `arr`? _______________________________
   b) How much memory is allocated for `arr2`? ______________________________
   c) Which of the above declared arrays can data be copied into without further initialization? Why?
   d) Suppose a sorting algorithm requires swap operations. Which of the above structures is more (time) efficient for swapping records? Why?

CS214 Fall 2017 Midterm Exam

Name: ______________________________ NetID: ________________________________

Scratch/Additional space: