UC Berkeley
Fall 2019
CS61C Quest

Q1) [10 Points] **Negate** the following **nibble *binary/hex*** numbers, or write N/A if not possible. Remember to write your answer in the appropriate base. (A nibble is 4 bits)

(Unsigned) 0b0101	(Bias = -7) 0b0100	(Bias = -7) 0xF	(Two's Comp) 0b1100	(Two's Comp) 0xA
0b	0b	0x	0b	0x

...scratch space below...

Q2) [6 Points] Which of the following sums will yield an **arithmetically incorrect result** when computed with **two's complement nibbles**?

Correct ◯ Incorrect ◯	Correct ◯ Incorrect ◯	Correct ◯ Incorrect ◯
0xD + 0xE + 0xF	0x7 + 0x8	0x3 + 0x5

...scratch space below...

Q3) [12 Points] For each of the following representations, what is the *fewest number of bits* needed to cover the given range, which is inclusive of the endpoints (e.g., [1, 4] is the numbers 1, 2, 3 and 4). Write "N/A" if it is impossible. For the **Bias *Value*** (final value = unsigned + bias value), we'll let YOU specify whatever offset you wish to minimize the total number of bits needed for the Bias encoding.

Range	Unsigned	One's Comp	Two's Comp	Sign&Mag	Bias	Bias *Value*
[0, 10]		5	5			0
[-4, -1]		4				
[1, 4]				4	2	

...scratch space below...

For this page, assume all `malloc`s are successful, all necessary libraries are `#included`, and any heap accesses outside what the program allocates is a segmentation fault.

Q4) [12 Points] Which of the following are possible, if perhaps unlikely, results of attempting to compile and run this code? (select ALL that apply)

```c
int main() {
    int32_t *str = (int32_t *) malloc(sizeof(int32_t) * 3);
    printf("%s", (char *) str); // A char is 8 bits.
    return 0;
}
```

☐ Compilation error due to invalid typecast

☐ Runtime typecasting error

☐ A segmentation fault

☐ The program prints the empty string

☐ The program prints **CS61C**

☐ The program prints **CS61C rocks!**

Q5) [10 Points] Each of the following evaluate to an address in memory. In other words, they "point" somewhere. Where in memory do they **point**?

	Code	Static	Stack	Heap
arr	○	○	○	○
arr[0]	○	○	○	○
dest	○	○	○	○
dest[0]	○	○	○	○
&arrPtr	○	○	○	○

Q6) [10 Points] The program below runs through the array of strings, doing something to each of the characters and putting the results in the `dest` array.

What are the first 8 characters the program prints? (Note: The program DOES compile and run without error.)

— — — — — — — —

```
// The ASCII values for 'A', 'B', etc. are 65, 66, ... ←←←←←← Important
// The ASCII values for 'a', 'b', etc. are 97, 98, ... ←←←←←← Important

char *arr[] = {"Go", "Bears"};

int main() {
    char **arrPtr = arr;
    char *dest[2];
    int j;

    for (int i = 0; i < 2; i++) {
        char *currString = *arrPtr;
        dest[i] = (char *) malloc(strlen(currString) + 1);
        for (j = 0; j < strlen(currString); j++) {
            dest[i][j] = currString[j] & ~(1 << 5); // Hint: Focus on this line!
        }
        dest[i][j] = '\0';
        arrPtr++;
    }
    printf("%s %s", dest[0], dest[1]);
}
```