CSM 61C
Spring 2020

Midterm 1 Review
Mentoring 1: February 12, 2020

1 Number Representation

1.1 What is the range of integers represented by a n-bit binary number? Your answers should include expressions that use 2^n.

(a) Unsigned:

(b) Two's Complement:

(c) Bias (with bias b):

1.2 How many unique integers can be represented in each case?

(a) Unsigned:

(b) Two's Complement:

(c) Bias (with bias b):

2 Memory Addresses

2.1 Consider the C code here, and assume the malloc call succeeds. Rank what each variable evaluates to from 1 to 5, with 1 being the least, right before bar returns. Use the memory layout from class; Treat all addresses as unsigned numbers.

```
#include <stdlib.h>

int FIVE = 5;

int bar(int x) {
    return x * x;
}

int main(int argc, char *argv[]) {
    int *foo = malloc(sizeof(int));
    if (foo) free(foo);
    bar(10); //  snapshot just before it returns
    return 0;
}
```

```
foo:   _______
&foo:  _______
FIVE:  _______
&FIVE: _______
&x:    _______
```

2.2 Consider the following C program:

```
int a = 5;
int main()
{
    int b = 0;
    char* s1 = "cs61c";
    char s2[] = "cs61c";
    char* c = malloc(sizeof(char) * 100);
    return 0;
};
```

For each of the following values, state the location in the memory layout where they are stored. Answer with code, static, heap, or stack.

(a) `s1`

(b) `s2`

(c) `s1[0]`

(d) `s2[0]`

(e) `c[0]`

(f) `a`

3 Linked Lists Revisited

3.1 Fill out the declaration of a singly linked linked-list node below.

```
typedef struct node {
    int value;
    ____________ next; // pointer to the next element
} sll_node;
```

3.2 Let’s convert the linked list to an array. Fill in the missing code.

```
int* to_array(sll_node *sll, int size) {
    int i = 0;
    int *arr = _________________;
    while (sll) {
        arr[i] = ___________;
        sll = ___________;
        _______;
    }
    return arr;
}
```

3.3 Finally, complete the function `delete_even()` that will delete every second element of the list. For example, given the lists below:

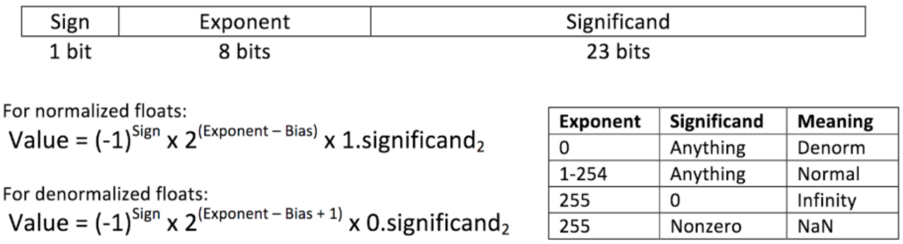
```
Before: Node 1 → Node 2 → Node 3 → Node 4
After: Node 1 → Node 3
```

Calling `delete_even()` on the list labeled ”Before” will change it into the list labeled ”After”. All list nodes were created via dynamic memory allocation.

```
void delete_even(sll_node *s11) {
    sll_node *temp;
    if (!sll || !sll->next) {
        return;
    }
    temp = ________;
    sll->next = ________;
    free(________);
    delete_even(________);
}
```

4 Floating Point Intro

The IEEE standard defines a binary representation for floating point using **sign**, **significant**, and **mantissa**.



Sign	Exponent	Significand
1 bit	8 bits	23 bits

For normalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \text{ x } 2^{(\text{Exponent} - \text{Bias})} \text{ x } 1.\text{significand}_2$$

For denormalized floats:

$$\text{Value} = (-1)^{\text{Sign}} \text{ x } 2^{(\text{Exponent} - \text{Bias} + 1)} \text{ x } 0.\text{significand}_2$$

Exponent	Significand	Meaning
0	Anything	Denorm
1-254	Anything	Normal
255	0	Infinity
255	Nonzero	NaN

4.1 (a) How would 10.625 be represented in floating point format?

(b) What decimal number is encoded as `0xC0A80000`?

(c) How many non-negative floats are strictly less than 2?

(d) What is the smallest positive value that can be stored using a single precision float?

5 RISC-V to C

5.1 Assume we have two arrays input and result. They are initialized as follows:

```
int *input = malloc(8*sizeof(int));
int *result = calloc(8, sizeof(int));
for (int i = 0; i < 8; i++) {
    input[i] = i;
}
```

You are given the following RISC-V code. Assume register x10 holds the address of input and register x12 holds the address of result.

```
    add x8, x0, 0
    addi x5, x0, 0
    addi x11, 0, 8
Loop:
    beq x5, x11, Done
    lw x6, 0(x10)
    add x8, x8, x6
    slli x7, x5, 2
    add x7, x7, x12
    sw x8, 0(x7)
    addi x5, x5, 1
    addi x10, x10, 4
    j Loop
Done:
    // exit
    . . . . . . . . . . . . .
```

```
// sizeof(int) == 4
int sum = 0;
```

5.2 What is the end array stored starting at register x12?