

## CSCI 2330 – Pointer Exercises

Consider a hypothetical machine with the following characteristics:

- 32-bit word size
- 16-bit int size
- 8-bit char size
- Big-endian

Suppose that you executed the following C code on such a machine, and that the compiler chose the indicated addresses to store each variable:

```
int i = 0x4321;    // stored at address 0x52
char c = 0xAB;    // stored at address 0x50
char* p = &c;     // stored at address 0x4B
```

1. Draw a diagram of memory from address 0x4A to address 0x55, showing the address and byte content (if known) at each byte in that range after running the above snippet of code. Write all addresses and values in hex (you can omit the 0x prefixes).

2. Repeat question (1) for a **little-endian** machine (all other characteristics the same).

3. What happens when the code snippet below is executed?

```
void do_something(int* p1, int* p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}
```

```
int x = 5;
int y = 3;
do_something(&x, &y);
```