Question 1.

1. 0x63333C49
2. -0x2 (0xFE was accepted to, even though I designed the question to make you enter a negative :p)
3. 0x3C49
4. 2^(C6-7F) * 1.333C49 or 1.667892   (or anything that turns into 3.3063112 * 10 ^ 21)
5. “I<3cs33”
6. Most reasonable sets were accepted (0,3,6), (1,4,7), (2,5,8)

Question 2.

1. No
2. Yes
3. Pretty much anything was okay here. I like little endian though. I think it’s not the case that x86 is weird because its little endian, it's that humans are weird for being big endian.

Question 3.

1. Func 1
2. Func 7
3. Func 2
4. Func 4

Question 4.

1. No
2. My_array + x*3*8 + y*8
3. No, because that would require multiple multiplication and addition operations to get the correct index

Question 5.

1. 16
2. Unsigned int
3. 0x400548
4.

Return addr
unused
0x4005bb
Old rbx
0x40055e

5. 0x400565

Question 6. It’s ackermann! (source https://tfetimes.com/c-ackermann-function/)

```
unsigned int ackermann(unsigned int m, unsigned int n) {
        if (m == 0) {
                return n + 1;
        }
        if (n == 0) {
                return ackermann(m - 1, 1);
        }
        return ackermann(m - 1, ackermann(m, n - 1));
}
```

Number of memory load instructions was not graded, because there were many good answers:
0 -- no explicit loads
3 -- pops and ret read the stack
All -- all instructions read memory because they are stored in memory

Question 7.

1. Indirect jump + switch case
2. several possibilities
    a. Any string where the first three characters contain 5’s and 0’s, but not a 6
    b. 460
3. 566