

## Lab 1 Problem 2

### Guess

1: a = 0022FF00, b = 0022FEFC, c = 0022FEF8, I = 0022FEF4

2: b = 002D0F98, c = FFFFFFFE

3: a[0] = 200, a[1] = 101, a[2] = 102, a[3] = 103

4: a[0] = 200, a[1] = 400, a[2] = 402, a[3] = 103

5: a[0] = 200, a[1] = 500, a[2] = 402, a[3] = 103

6: a[0] = 200, a[1] = 500, a[2] = 800, a[3] = 103

7: b = 0022FF01, c = 0022FF02

### Output

1: a = 0x7fff977bc260, b = 0x7fff977bc258, c = 0x7fff977bc250, I = 0x7fff977bc24c

2: b = 0x1db6010, c = 0x7fff977bc376

3: a[0] = 200, a[1] = 101, a[2] = 102, a[3] = 103

4: a[0] = 200, a[1] = 400, a[2] = 402, a[3] = 404

5: a[0] = 200, a[1] = 500, a[2] = 402, a[3] = 404

6: a[0] = 200, a[1] = 205044, a[2] = 256, a[3] = 404

7: b = 0x7fff977bc264, c = 0x7fff977bc261

- 4: I didn't recognize that 3[c] was equivalent to c[3]
- 6: I'm unsure how casting the values at a[1] and a[2] came about with the two declarations:
  - c = (int \*) ((char \*) c + 1);
  - \*c = 800;
- 7: It appears that casting as an int made b move 4 bytes up while casting as a char and then an int moved c took the original address and added 1 to it.