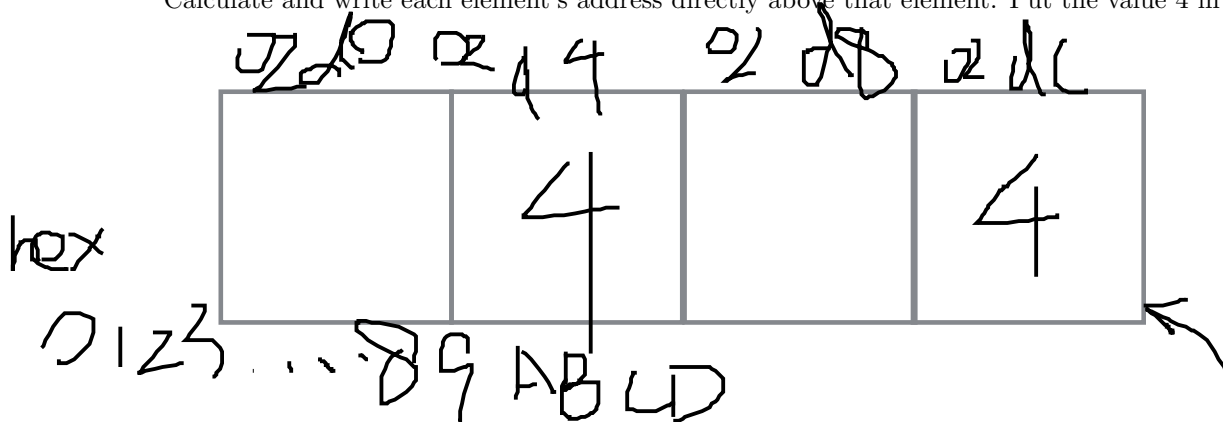


## CSC209H Worksheet: Arrays

- The array pictured below holds 4 integers and starts at memory address 0x00007ffcc2a502d0. It was declared as `int house[4]`; and the size of an integer on this machine is 4 bytes.

Calculate and write each element's address directly above that element. Put the value 4 in `house[1]`.

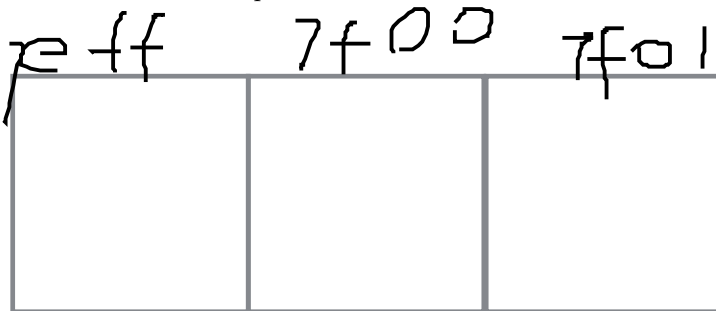


- On the array above, show the results of executing the statement: `house[3] = house[1];`

- This next array picture has 3 elements, but each element is a char.

It was declared as `char course_prefix[3] = {'c', 's', 'c'};`

The address of the middle element is 0x00007ffdf05c7f00. Calculate the addresses of the other two elements and write them on the picture.



- Write a C program that declares an array `ages` with 4 elements initialized to 5, 7, 18, and 20. In your program, use a loop to add 1 to each element. Then in a second loop, print out each element's index and its value with messages something like, "The element at index 0 now has the value 6".

Find someone near you with a computer. Compile and run your program from the command line.

- Add the following line to your program: `ages[4] = 99;` What is wrong with doing that? When you run your program, do you see any difference?
- Play around with changing the value of 4 to get an address farther away from your array. What happens when you change it to 100? What happens when you change it to -1? How large (or small) can you set that value before your program crashes?