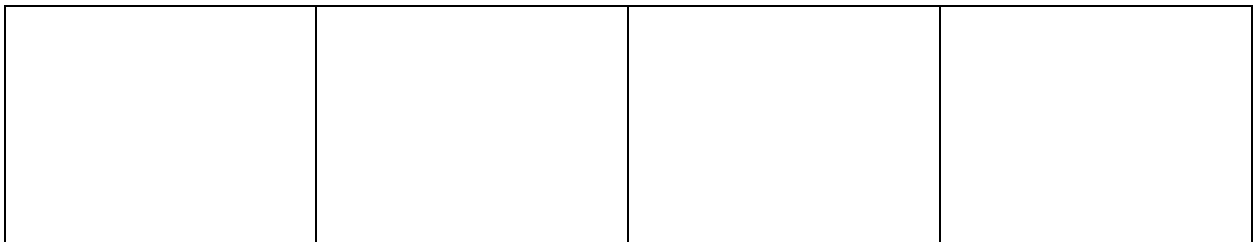


Week 3 in-class exercises. Pointers

Exercise 1. 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];`

Exercise 2. 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.



Exercise 3. Complete the following program by correctly implementing and calling function *swap* which swaps values in 2 integer variables:

```
#include <stdio.h>

void swap ( , ) {

}

int main() {
    int a=1, b=2;
    printf("a=%d,b=%d\n", a, b);

    swap ( , );
    printf("After swap a=%d,b=%d", a, b);
    return 0;
}
```

Exercise 4. Trace through the following program by hand to see why lying about your age doesn't last. Fix the program by changing the signature of the function to use a pointer.

```
#include <stdio.h>

void lie(int age) {
    printf("You were %d years old\n", age);
    age += 1;
    printf("You are now %d years old\n", age);
}

int main() {
    int age = 18;
    lie ( age );
    printf("But your age is still %d\n", age);
    return 0;
}
```

Exercise 5. Implement function *statistics* which takes as an input an integer array and the size of this array, and computes min, max and average of the elements in the array. How would you make the calling function to get all 3 values back?