scanf's behavior with whitespace

- Check the `read_char.c` program. For each way of reading the input, try to enter just a character and in the next run of the program, try to enter a space followed by a character and see how the program behaves. Note how the newline left from the first read of scanf will get read by getchar. If you want to see the behavior of each case separately, comment out the other two.

Element out of bounds

- Check `bounds.c` to see what happens when you access an element beyond the bounds of the array

Find all repeated digits

- Check `find_repeated.c` for a solution

Randomly dealing a hand of cards

- Check `deal_hand.c` for a solution of the problem. See more about the `rand()` function: https://www.tutorialspoint.com/cstandardlibrary/cfunctionrand.htm. Given that the rand function returns a number between 0 and `MAX_INT` (which is the maximum integer that can be represented), then if we want to have a random number generated from 0 to n-1, we just get the remainder of the returned value after dividing by n. For example, assume we want to generate a random number in the range [0,4) -- i.e., 0,1,2,3 -- then if we do `THE_RANDOM_NUMBER % 4` , the returned remainder would either be 0, 1, 2, or 3 which is what we want.

Example programs not covered in class

- Check `array.c` for different ways of initializing a 1-d array
- Check `two-d-array.c` for an example of using a 2D array. Note that this is a float array.
- Check `var-length-array.c` for an example of using a variable length 1-d array. Note that it will depend on your computer and compiler if you get some garbage values or not. REMEMBER YOU MUST NEVER ASSUME THE ARRAY IS INITIALIZED TO 0 UNLESS YOU DO IT YOURSELF.
- Check `two-d-var-length.c` to see how you can create variable length 2d arrays