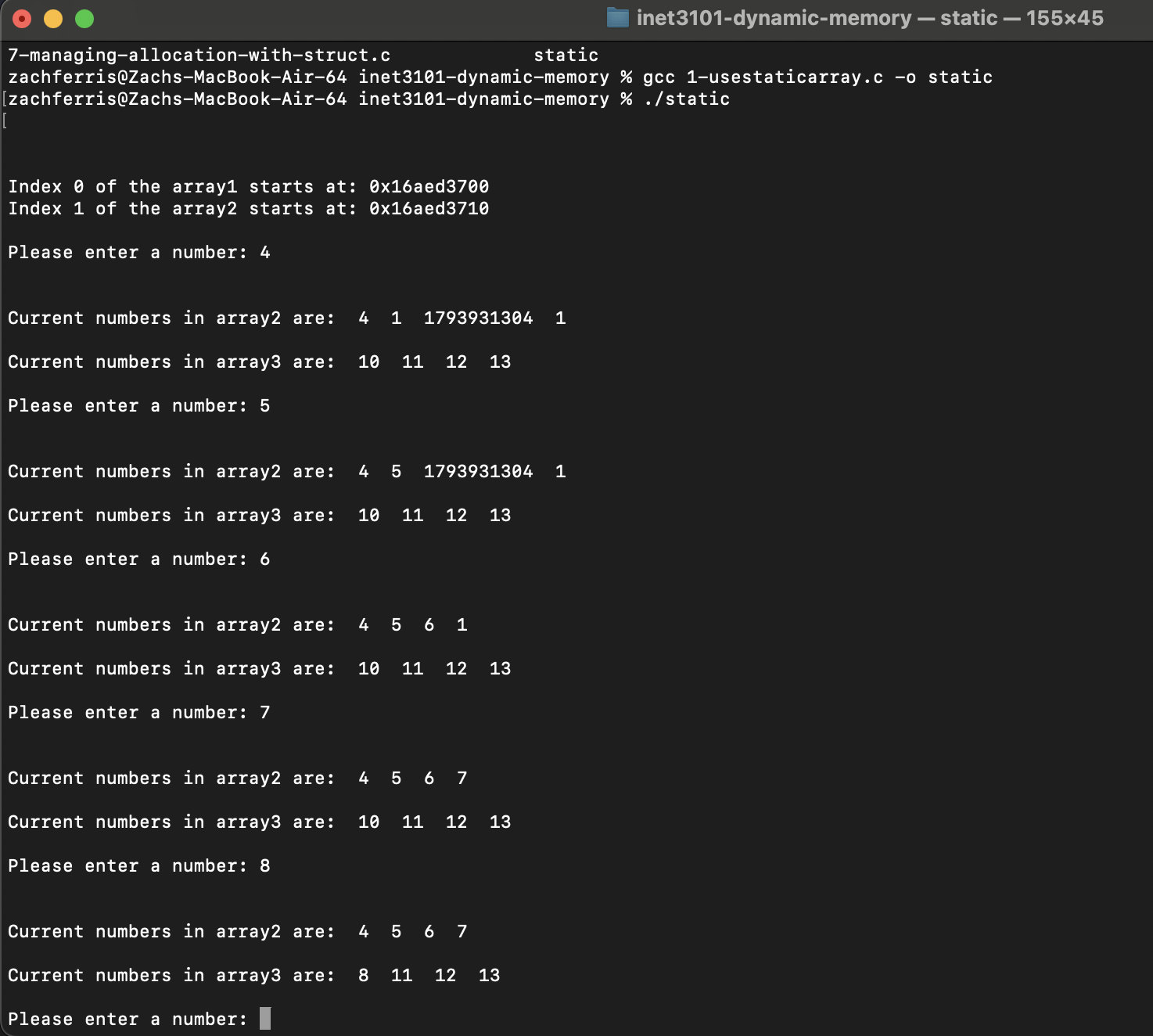
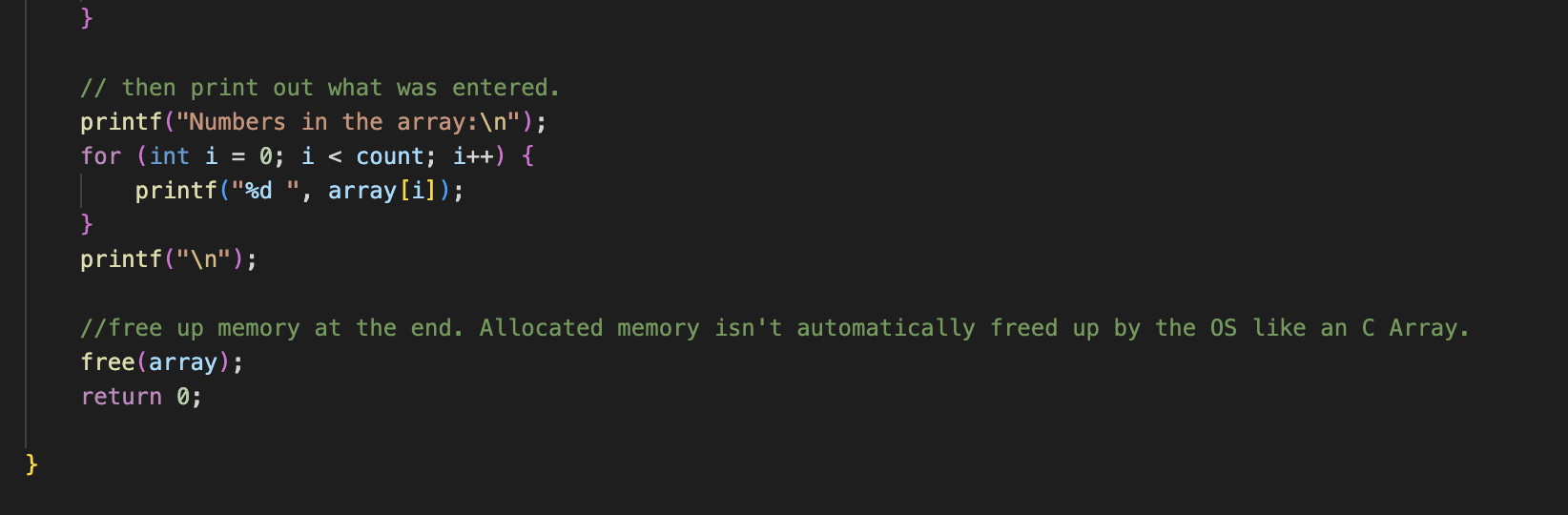
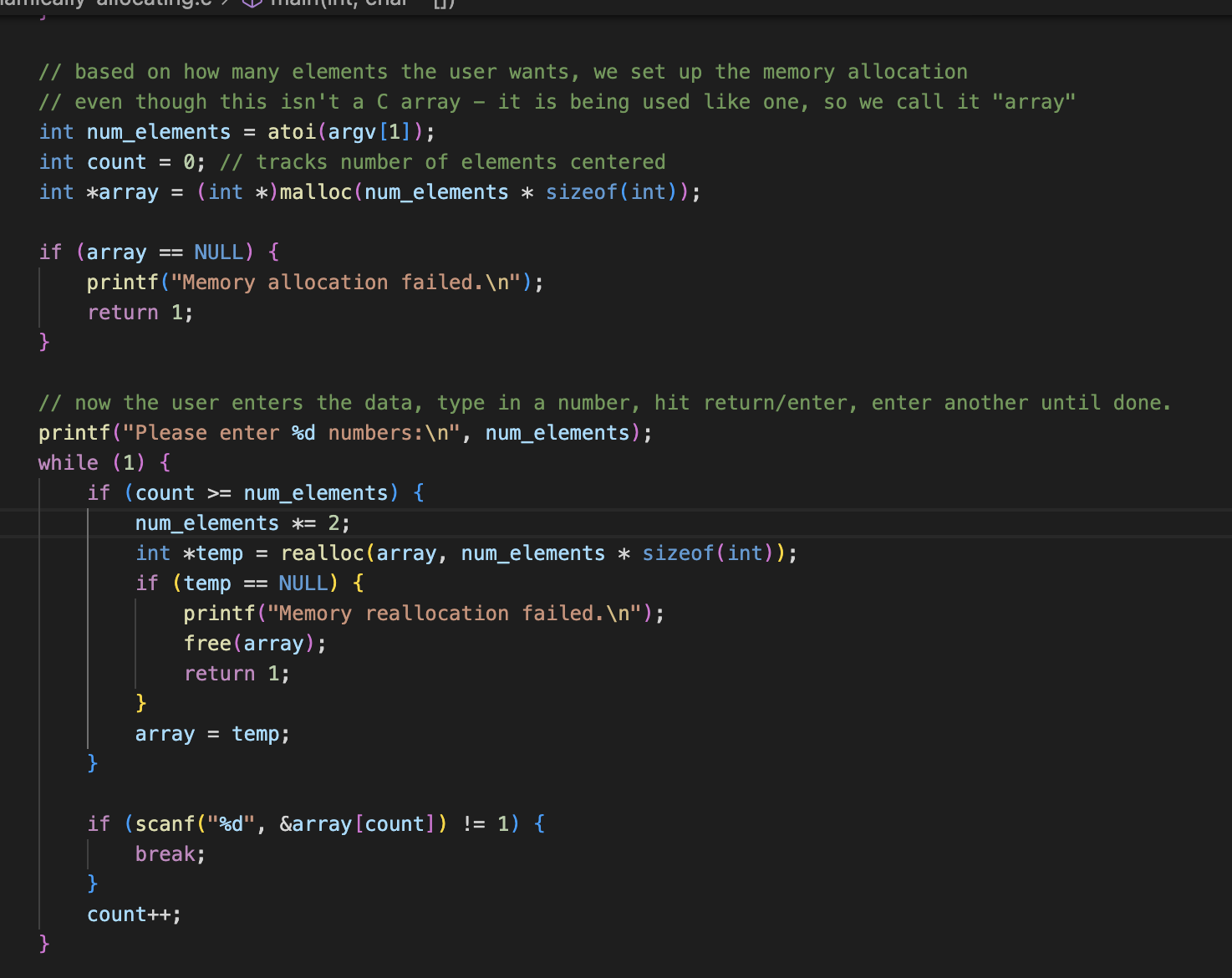
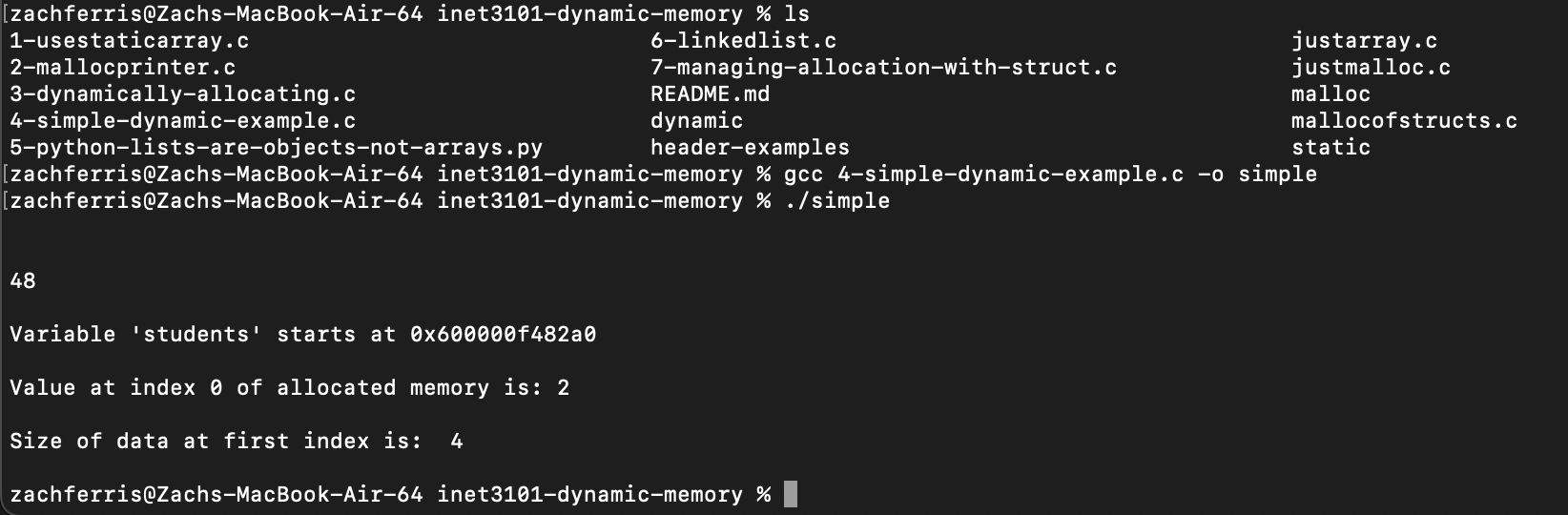
1.

2. N/A

3. Replacing the **for** loop with a **while** loop allows the user to continuously add more inputs, rather than just having a fixed amount of memory. Using realloc(), the program will double the allocated memory to make sure that there is enough space for new inputs. When using realloc(), it was making extra zeros be printed with the for loop because it was printing the entire allocated array even if some of the memory wasn’t being used. I added the count variable to be used instead, which tracks the number of valid inputs. Finally, users can now enter a non-integer to break the while loop, using scanf. The code that was used expects an integer, so if it is anything other than an integer, the loop will break rather than entering an infinite loop, making it much cleaner.





4. 

5. In object-oriented programming, an object is something that stores data (attributes) and has actions (methods) that can be performed on it. Objects are created from classes, which then act like blueprints. An example is, a car object might have different attributes like brand and model, and then have methods like drive(). In Python, a list is also an object because it stores data and has methods like .append() to add items and .sort() to sort/arrange them. This shows how objects can combine data and behavior to make code more organized and reusable.

6. Using a linked list allows us to store and manage data without needing a set size. An array needs a set amount of memory when created and a linked list grows and shrinks as needed by adding or removing nodes. This prevents wasted space and avoids the need to resize memory like we do with realloc(). Each node in a linked list stores data and a link to the next node, which makes it easy to insert or remove items without the need to shift elements, making it a big advantage over arrays.

7. N/A