a) Stack grows up, if I call b = mallac(16); then printf(“b = %p, &b %p”, b, &b); the second address, &b, will be larger than b which means my stack is smaller than the heap therefore the stack is growing up.
b) 4.2.1 based on Apple Inc build 5658
c) The malloc call is getting a higher memory address than the calls later in the code
The compiler is optimizing the calls and arranging the memory differently from how it is getting called, for example b is 0x7fff5fbff7d8 while c is next at 0x7fff5fbff7d0 but it should be further away
d) a = 0x7fff5fbff7e0, b = 0x7fff5fbff7d8, c = 0x7fff5fbff7d0, i = 0x7fff5fbff7cc - this comes from executing the program and capturing the output
e) b and c are both pointers and therefore have the contents of a memory address pointing at nothing
f) 16 bytes