1.
a: offset 0, size 2
b: offset 4, size 4; we need 2 bytes of padding so that the int starts on a 4-byte boundary and can be read in one instruction
c: offset 8, size 1
d: offset 12, size 8; 3 bytes of padding are needed to start on 4-byte array to read the double in 2 instructions
total size: 20

2.
a: offset 0, size 2
b: offset 2, size 4
c: offset 6, size 1
d: offset 7, size 8
total size: 15
Since our goal is as little memory usage as possible, and the lowest addressable size in our architecture is 1 byte, we can use no paddings and start a variable on any address, even though it will take more instructions to fetch its value.