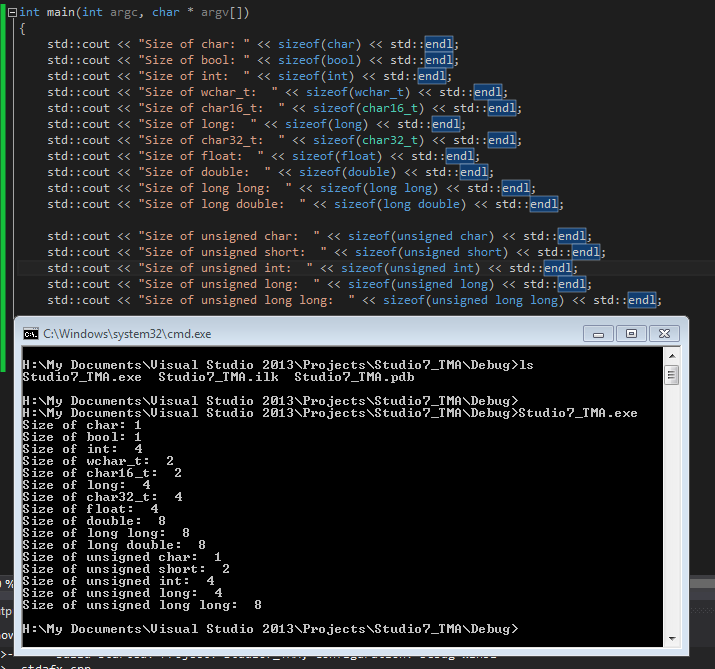
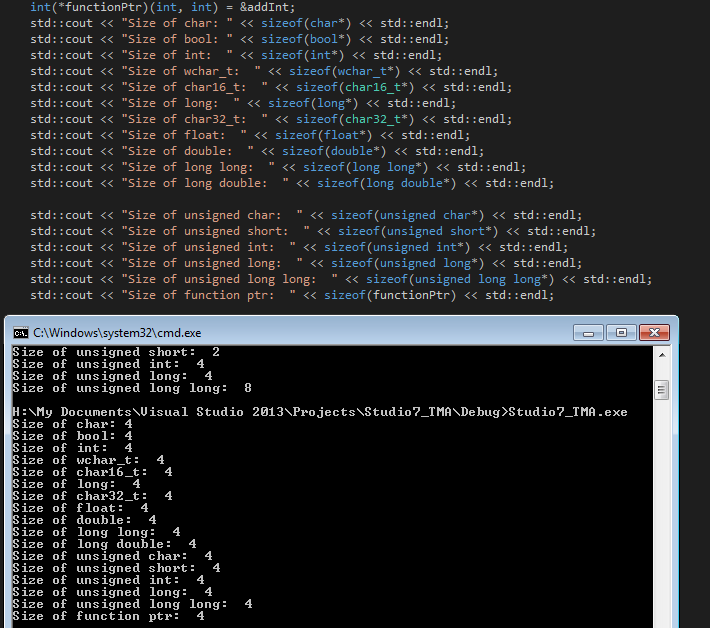
1. Christopher Ogle, Peiyun(Seed) Zeng, Menga Lin



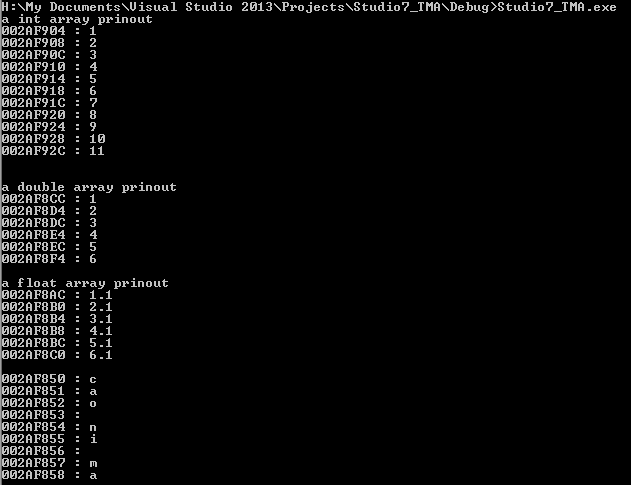
Yes, after checking the slide and comparing against our result all rules were obeyed.

3.



For this we notice that all the pointers are of size 4 bytes. Which makes sense since a pointer will be holding an address. Although our machine is 64bits, each address is 32 bits.

4.



We see the relationship that we would expect.

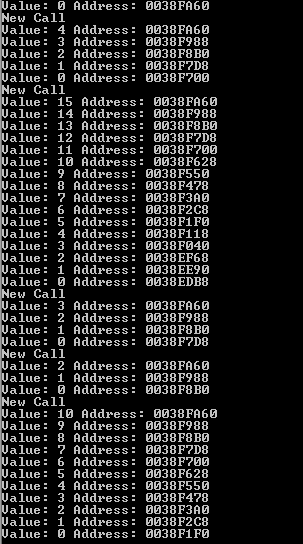
When we have an array of ints we see after each incrementation that the pointer moves by 4 bytes, this makes sense since the C++ standard defines an integer to take up 4 bytes.

When we have a double we see that the value of the pointer moves by 8 bytes. This again makes sense since a double occupies 8 bytes in memory.

When we have a float we see that the memory increases by 4 bytes since it is specified in the C++ Standard to be of size 4 bytes.

Finally the characters increment by one byte. This is because the characters are of size one byte, thus to access the next element we must increment it by a single byte.

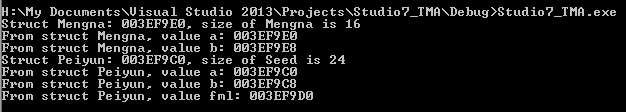
5.



(1)Every recursive call initialize a new stack frame that is further down in memory. Thus, the unsinged int that is on the stack from is also further down

(2) /(3) : yes, The

6.



So we realize that Struct’s size is the sum of the sizes of all member variables Plus necessary paddings.