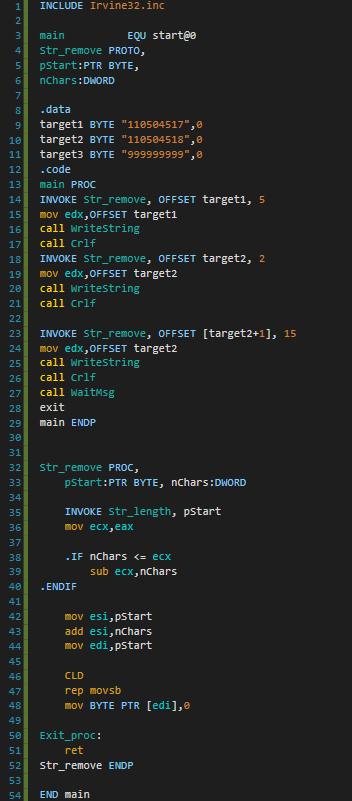
**Week 9 Lab Assembler - StringRemove**

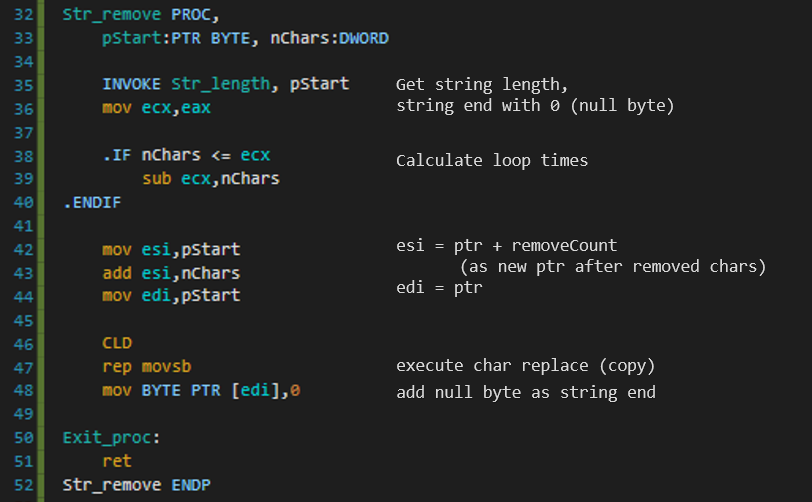
**Group 65**

**Integrants: 110504517 李睿穎 (Leader)，110504518 鍾秉均 (Member)**

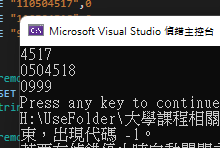
**Full Code**

****

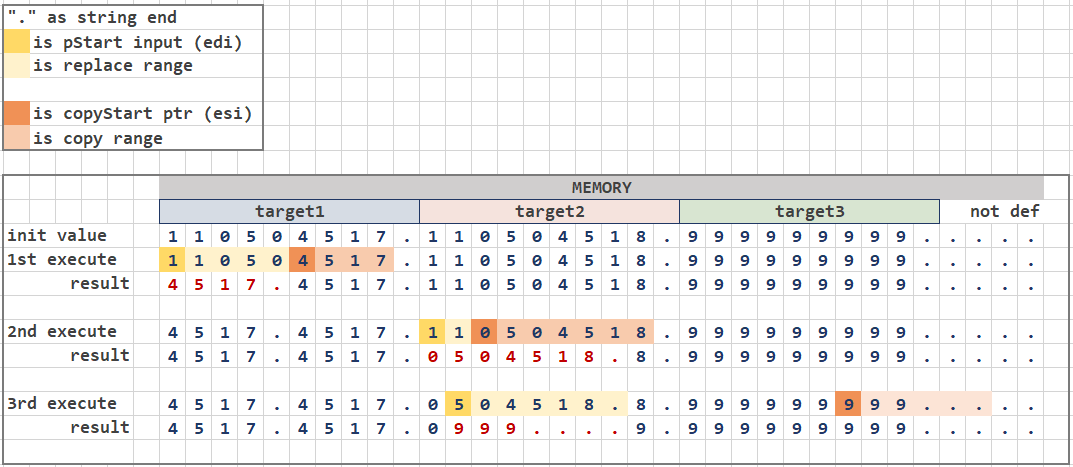
**Explanation**

****

**Print Result**

****

**Reason of 3rd Execute result**

****

When executing Proc StringRemove, the program directly executes it and doesn’t do a length check, this means that when the program surpasses the length of the input the memory just does an adjustment with what is left. In the third execution, when the program removes 15 characters, it has already surpassed the input string length (6) so it does an adjustment. In the proc there isn't any surpass input String len case, so what it does is copying and not removing.

**Reflection**

This part of the lab was a little hard to understand because we weren't sure how the operators and function worked.  We were a little confused by the result it gave us and how the entire process worked. At first we didn't notice that when we deleted the string it would copy what what left in the places that we just deleted. We also didn't know where assembler starts to count the bits that he need to delete. Basically we were a little confused by the result and finally discovered how it worked.

In conclusion it was a interesting experience were we tested how things like: rep, movsb, str\_remove worked.