Assembly Language and System Programming Lab6 Report

Group: 2

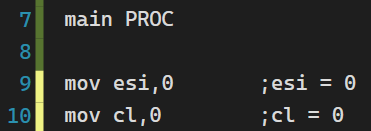
Names: 林柏廷(Brian)、蔡淵丞(Vincent)

Student ID: 110502531、110502567

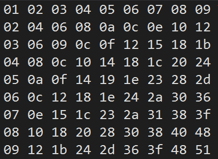
**Objective:**

Create a multiplication table from 1\*1 to 9\*9 and save the result.

**main PROC:**

****

Use ***esi*** as an index of the multiplication table. ***cl*** stores the number of row that we’re working on, starting from 0 since once we enter L1, we would increment it by one.

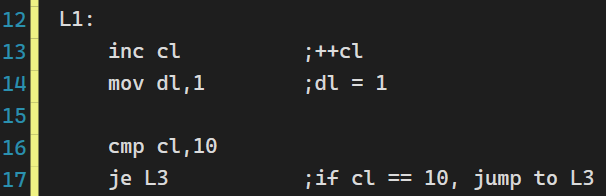
number of rows 

**L1:**

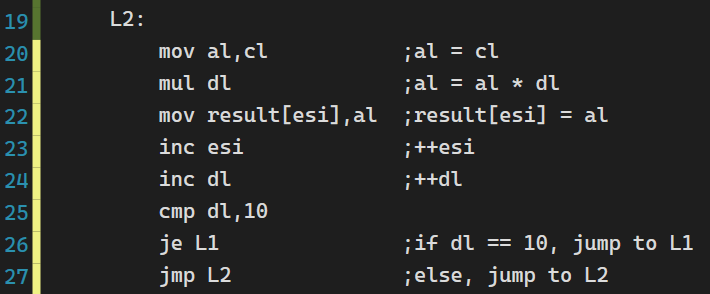
When processing each row, we would enter L1 first. Since ***cl*** would store the last row processed, we first increment ***cl*** by 1.

We use ***dl*** as the multiplier, and since the row numbers are multiplied by numbers from 1 to 9, we first store 1 into ***dl***.

Compare ***cl*** with 10, if they’re equal, jump to L3 and exit, since that would mean we have finished all the 9 rows.

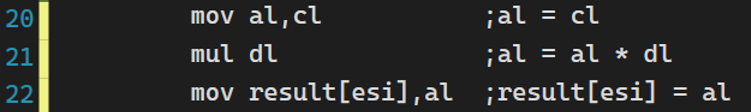
****

**L2:**

****

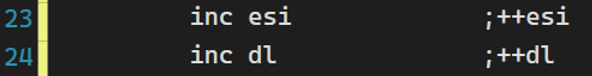
**Line 20 – 22:**

To multiply ***cl***, we first store it in ***al***, then use the mul operator to multiply it by ***dl***. After that we can store it to the result.

****

**Line 23 – 24:**

Increase ***esi*** by one to reach to the next memory location of result. Also, we increase ***dl*** to get the next multiplier.

****

**Line 25 – 27:**

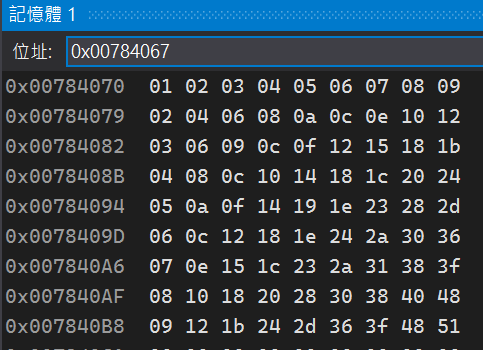
Compare ***dl*** to 10, if they’re equal, jump back to L1 and start working on the next row number. Since if ***dl*** equals to 10, we’ve already multiplied current row by 1 to 9.

If ***dl*** and 10 are not equal, jump to L2 and start multiplying current row by the next number.

****

**L3:** exit.

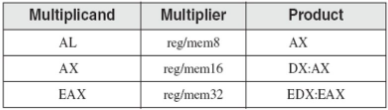
**Result:**

****

**Review:**

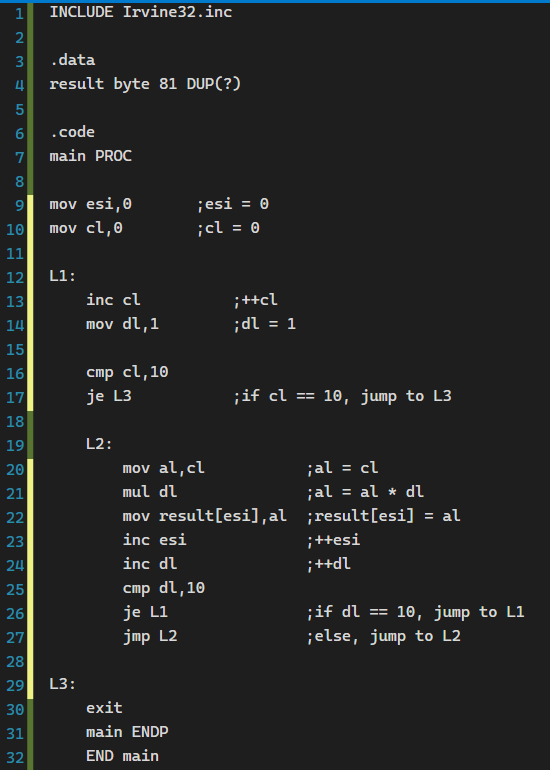
There are different approaches to this problem. Instead of using nested loop, we use compare and conditional jump operators, they’re basically similar to what loops do.

In this lab we’ve learned about what multiplication instructions do. We need to be extra careful with the size since different multipliers would lead to different multiplicands and products as the following picture shows.



However, since we’re only creating a multiplication table up to 9\*9 in this lab, a byte would be enough.

**Full code:**

****