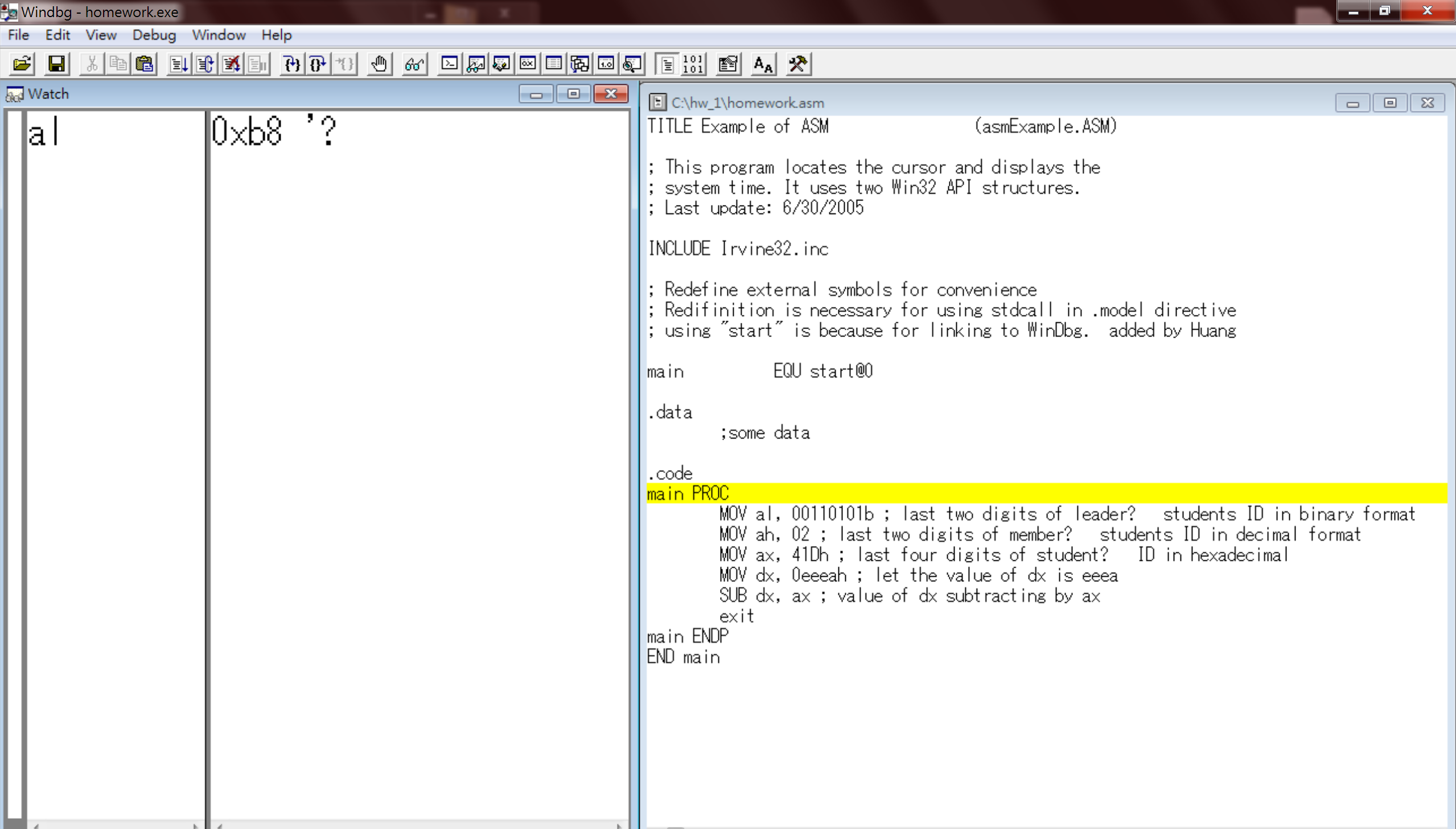
**In-Class Exercise # 2 – “Start Assembly Programming”**

**Group:** 61

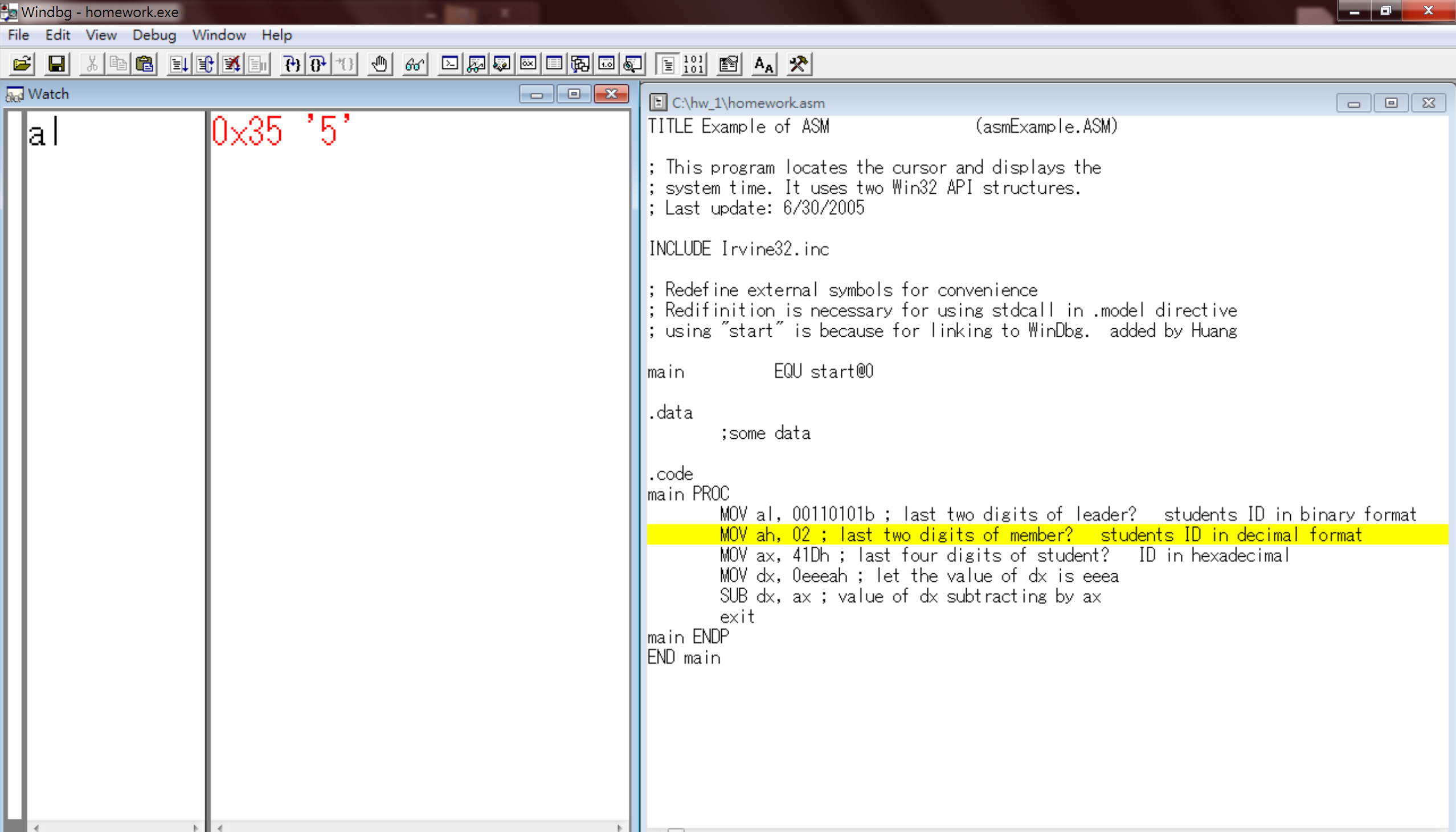
**Name & ID:** 陳宥涵 (Joy) 102401053 趙慈暄(Bonnie) 104503002

**Program execution flow**

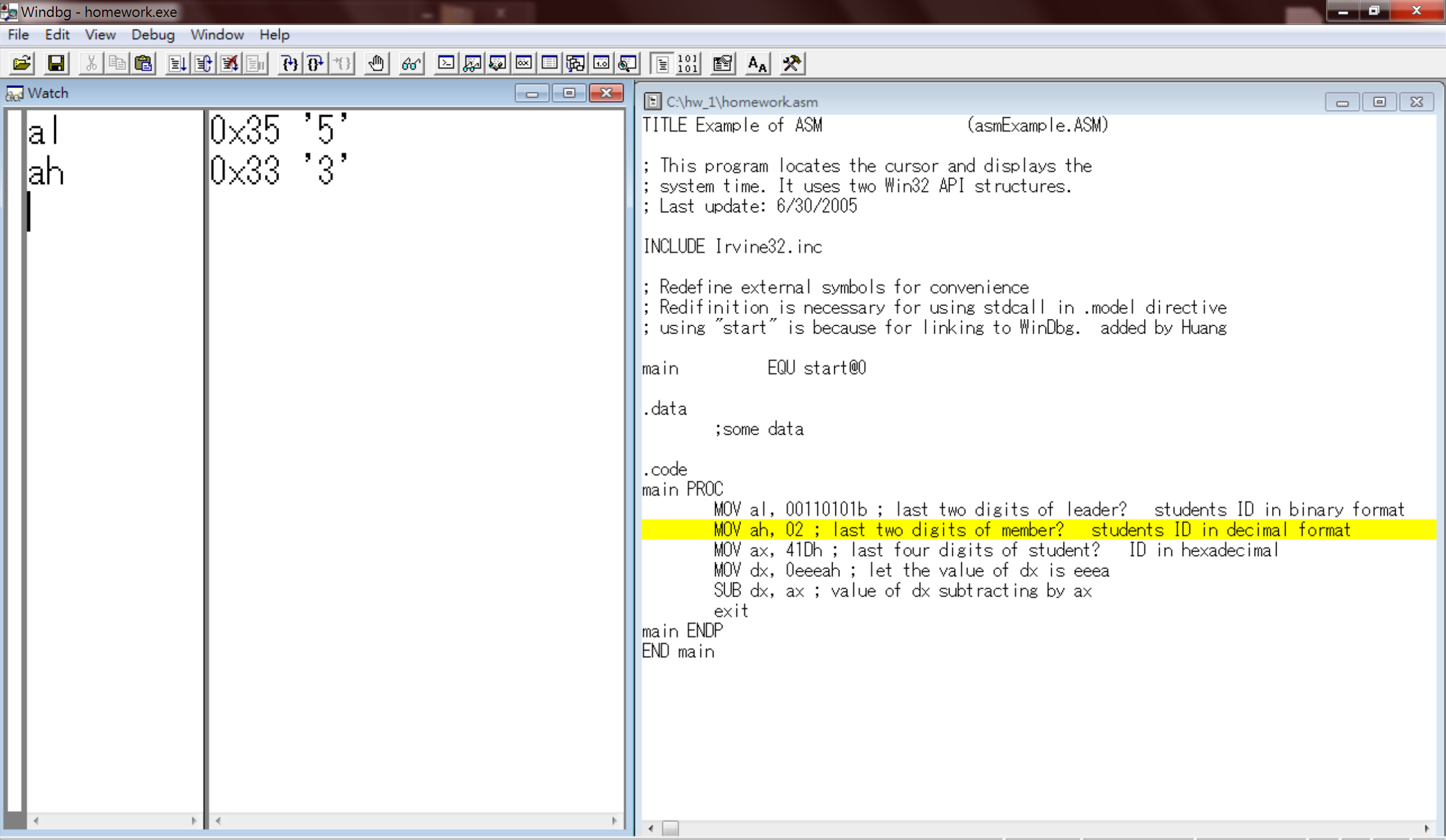
Before executing the first line of the code, first take a look at the al register on the left screen. The current value is 0xb8.



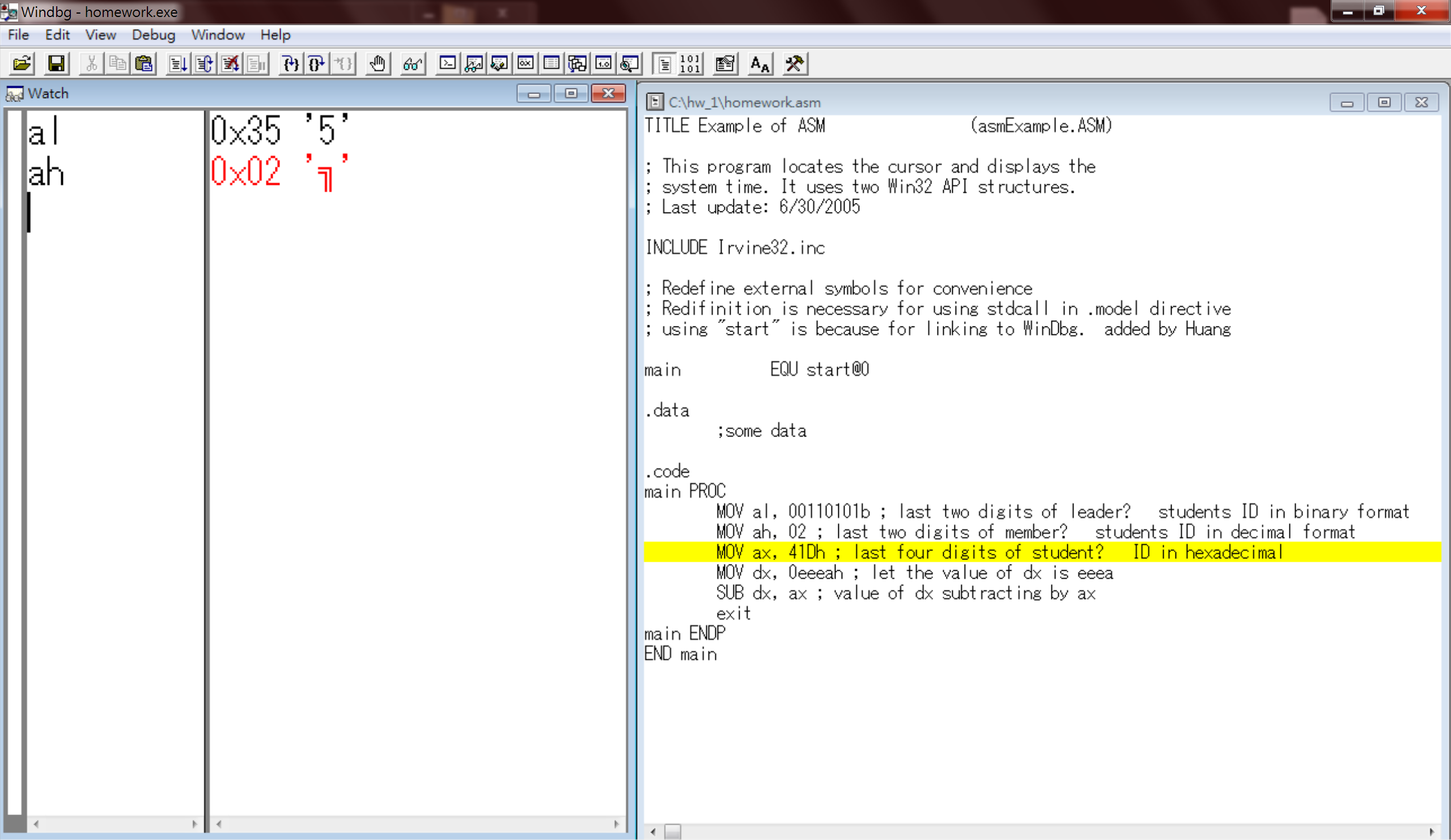
After executing the first line to move the last two digits of leader’s student id to register al, from the left screen, we can tell that the value became 0x35, and it’s correct because the hexadecimal value of 53 in decimal is 35. Plus, we have to add a ‘b’ after the binary format of 53.



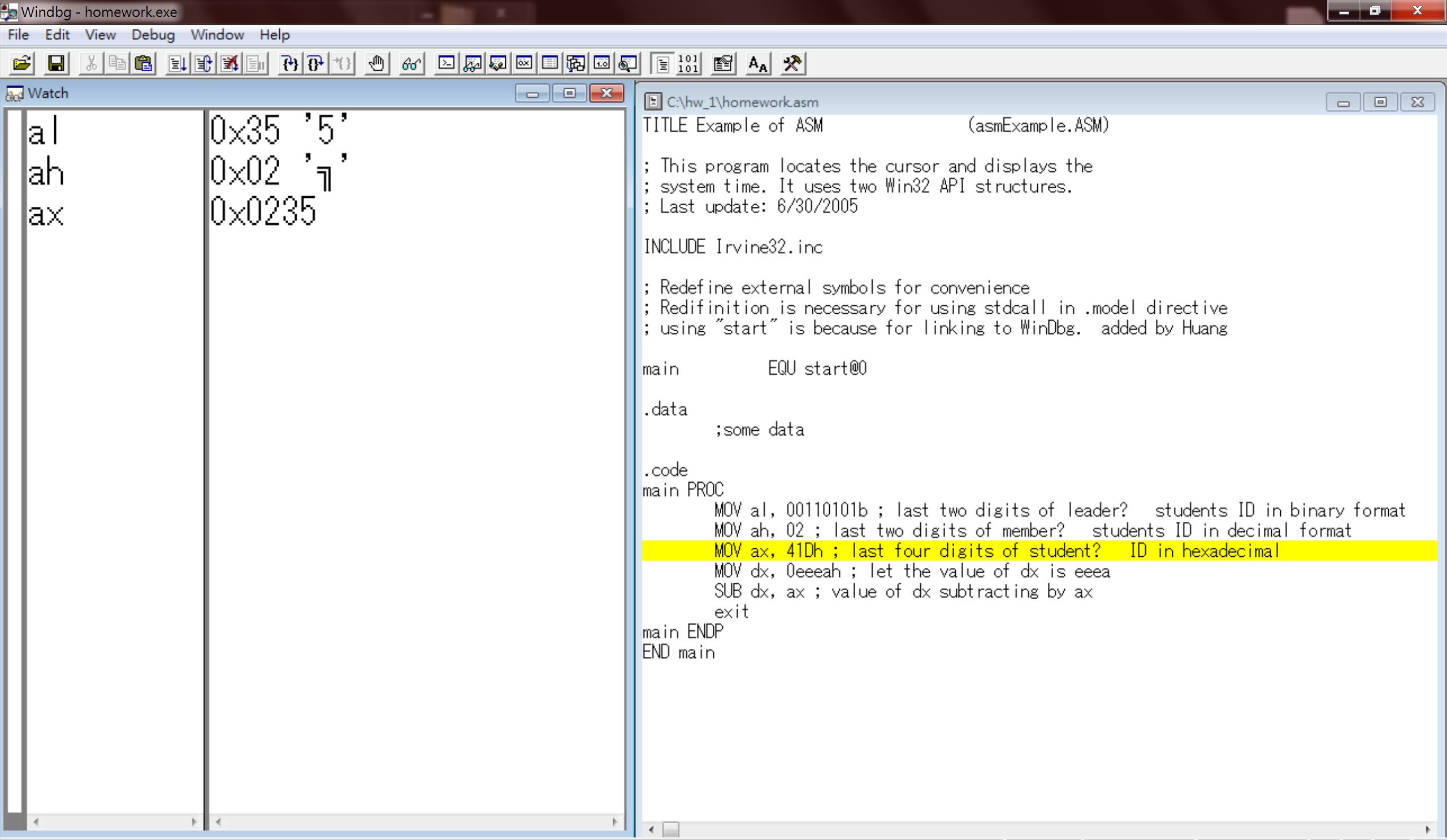
Before executing the second line of the code, we take a look at the initial value of ah register, the result is in the left screen. The value of the ah is 0x33.



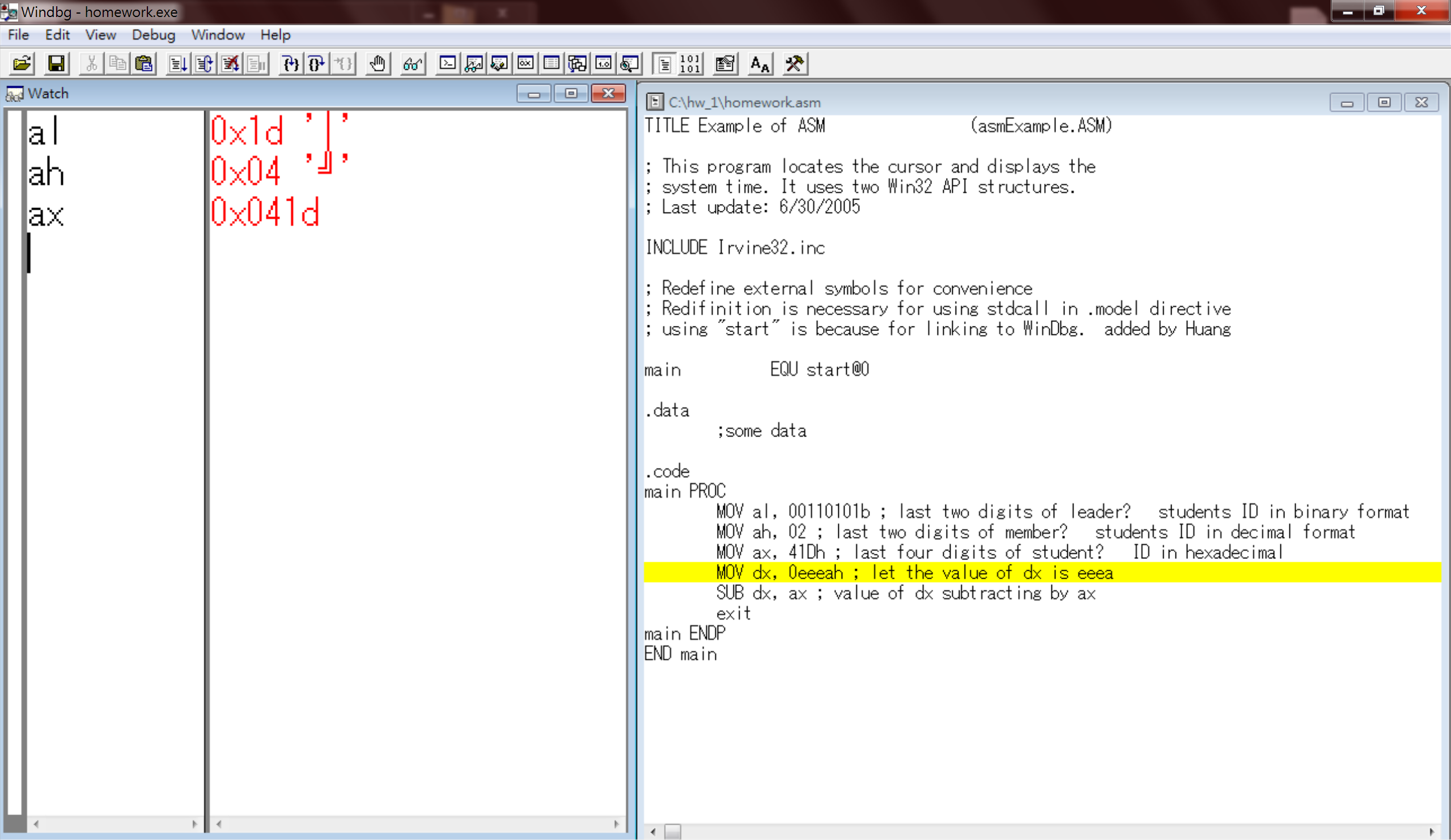
After moving 02 to register ah, we can now see the value of register ah became 0x02, which 02 is correctly the hexadecimal value of decimal 02.



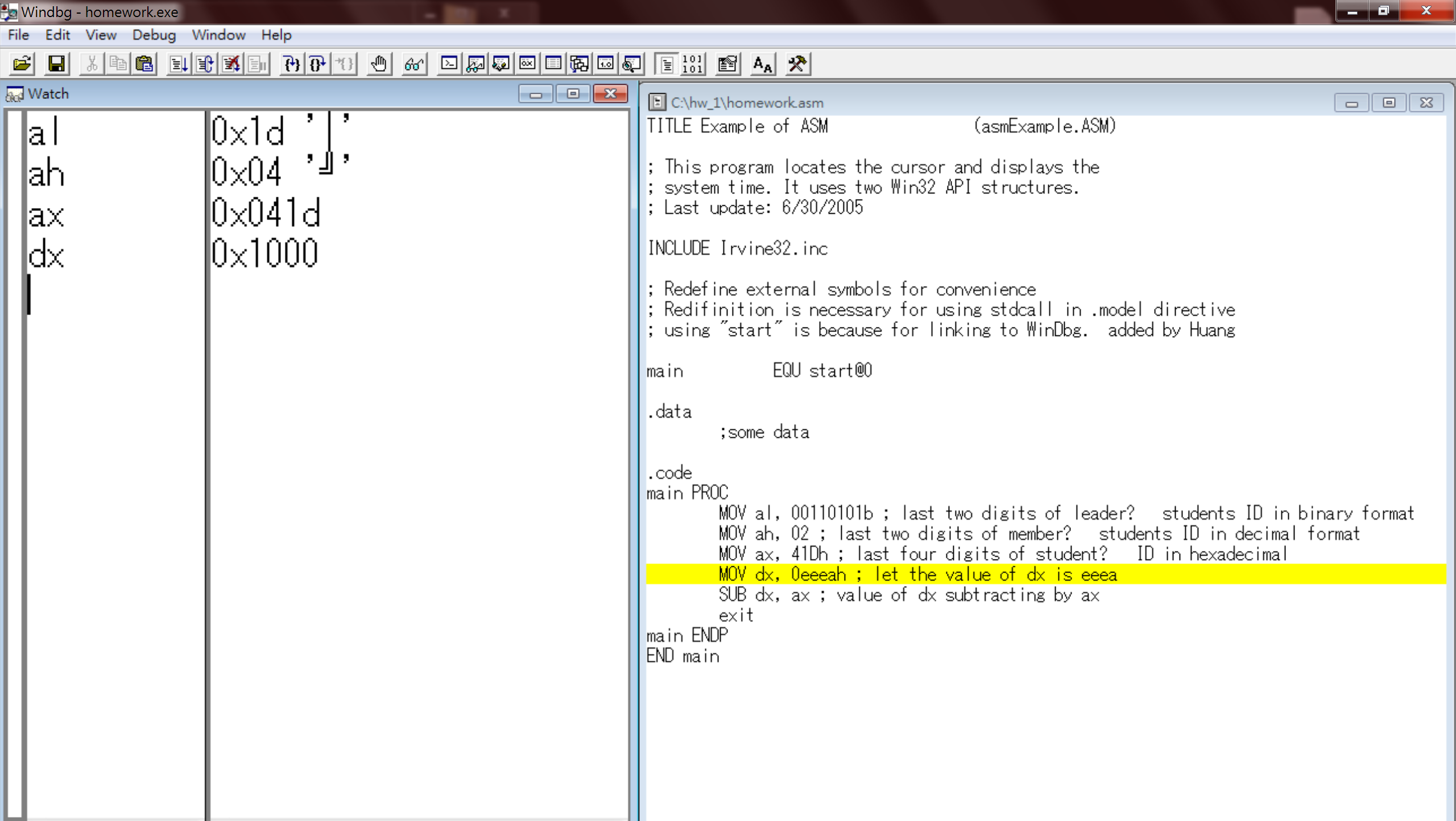
From the left screen, we can tell the initial value of ax is 0x0235.



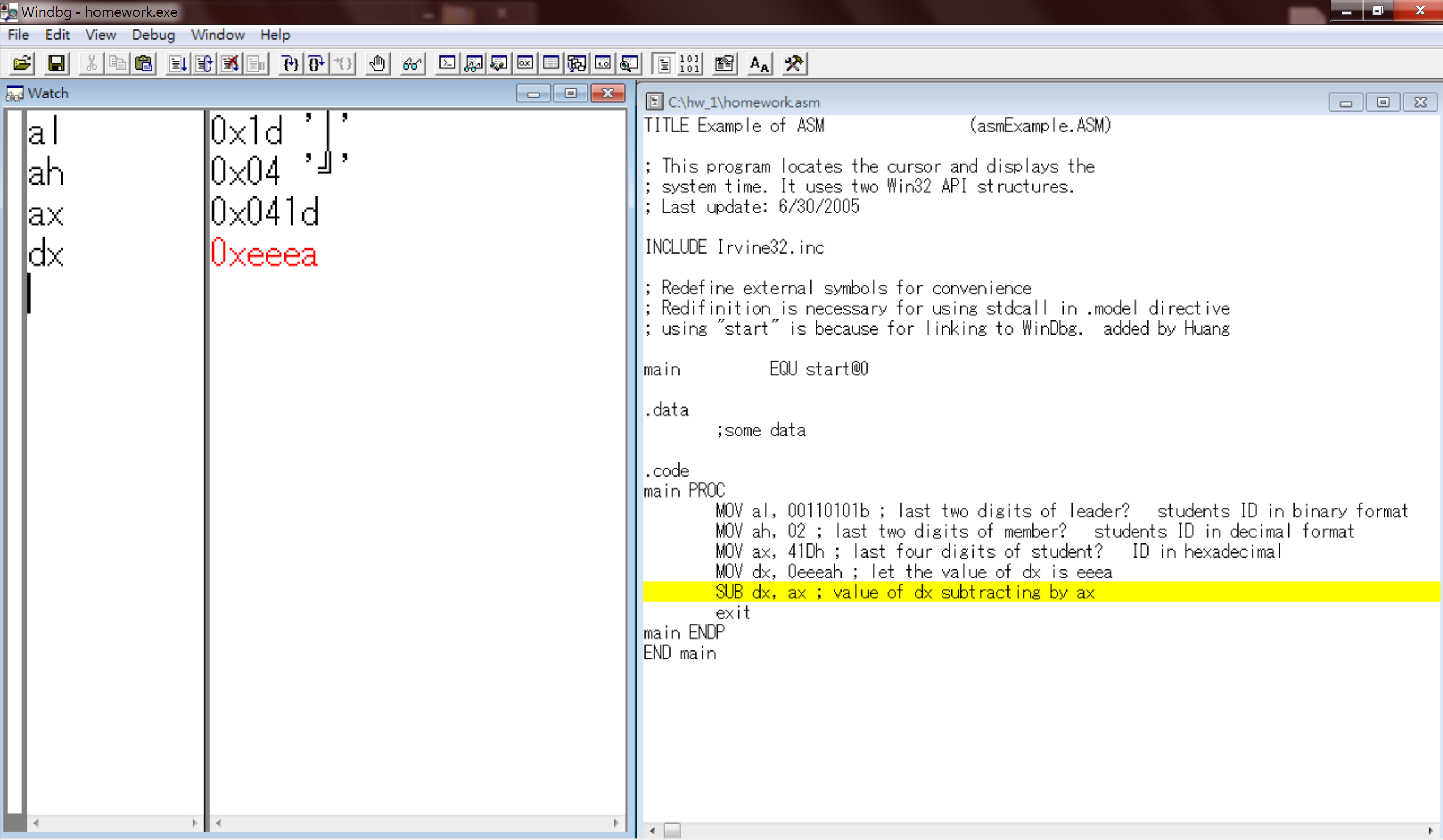
After moving 41d to ax, the value became 0x041d. 41d is the hexadecimal value of 1053, and for this line of code, we need to remember to add a ‘h’ after 41d to inform the assembler that it’s in a hexadecimal format. We can tell from the left screen that we’re getting the correct value.



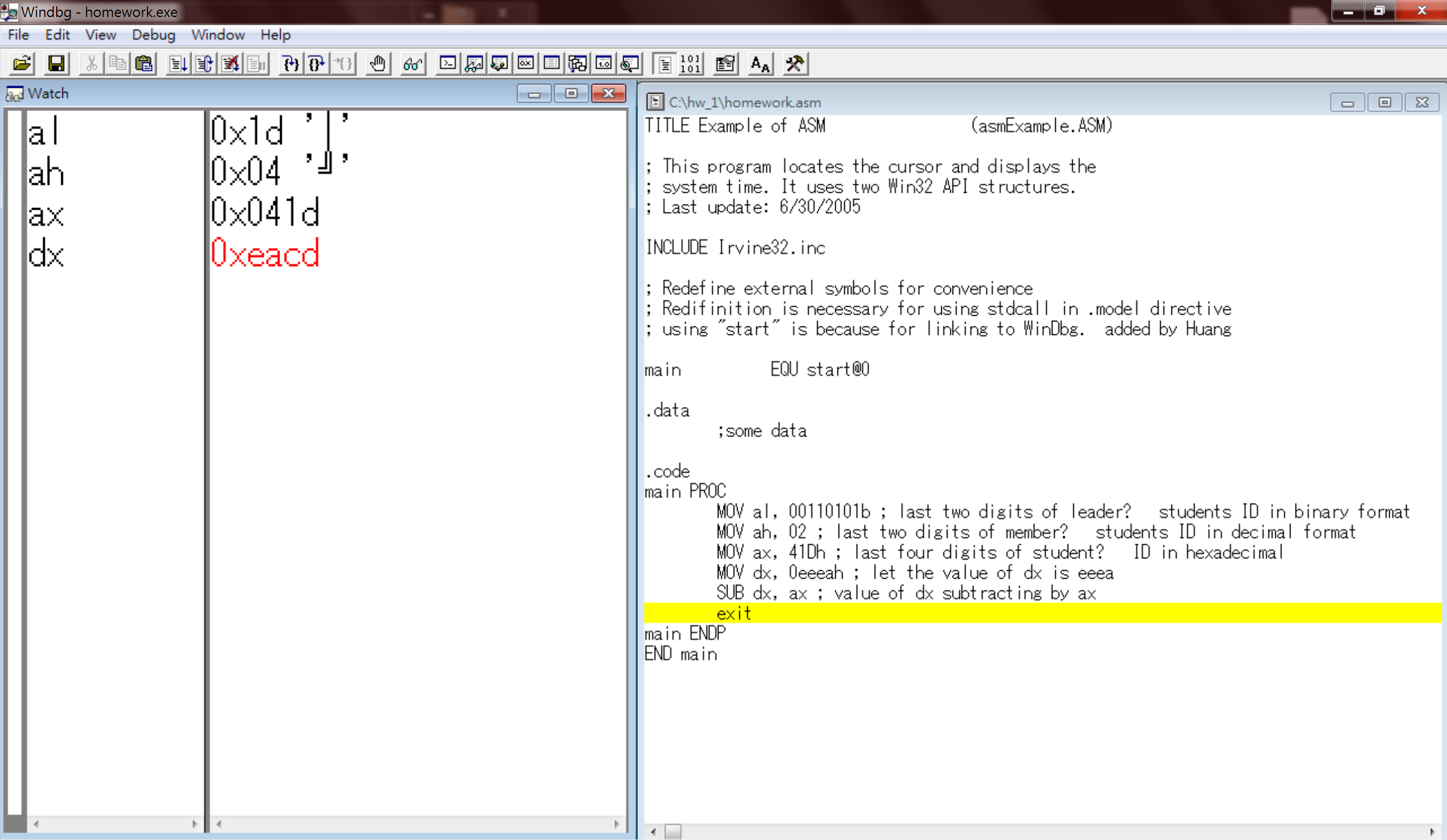
The initial value of dx is 0x1000, later on, we’re going to assign eeea to it.



We can see that after the line of code is executed, the value of dx register became 0xeeea, which corresponds to what we intended to store in it.



Last, we’re going to subtract dx by ax, which means the operation dx – ax, and by calculating by hands, we expect a value of 0xeacd which corresponds to the value on the left screen, so the value of dx register is correct.



**Reviews:**

Today is our first hands-on experience on assembly language, me and my teammate are both excited. Wanting to implement what we've learnt from the lecture and to confirm some confusing concepts by actually executing the code. For instance, whether it's necessary to add a 'd' for decimals, now we confirmed that it is not a mandatory for decimals. Another example is that if we forget to add a 0 in front of hexadecimal value which starts with an alphabet, an error will truly occur. It really excites us of being able to substantiate some of the concepts gained during the lecture.

Frankly, it's pretty blur in the beginning of the lab session since it's the first time for us to code in assembly language, so that we can catch up with the TA;

however, I found that by going through the slides provided by TAs, it's helpful for us to understand. TAs are super patient and nice, whenever we had problem, they're willing to accompany us and help us through the problems. Thanks to TAs, we've had a great experience on today's first lab session and continue to possess great enthusiasm in assembly language.