National University
Of Computer and Emerging Sciences

LAB 07:

Name – **Muhammad Taha**

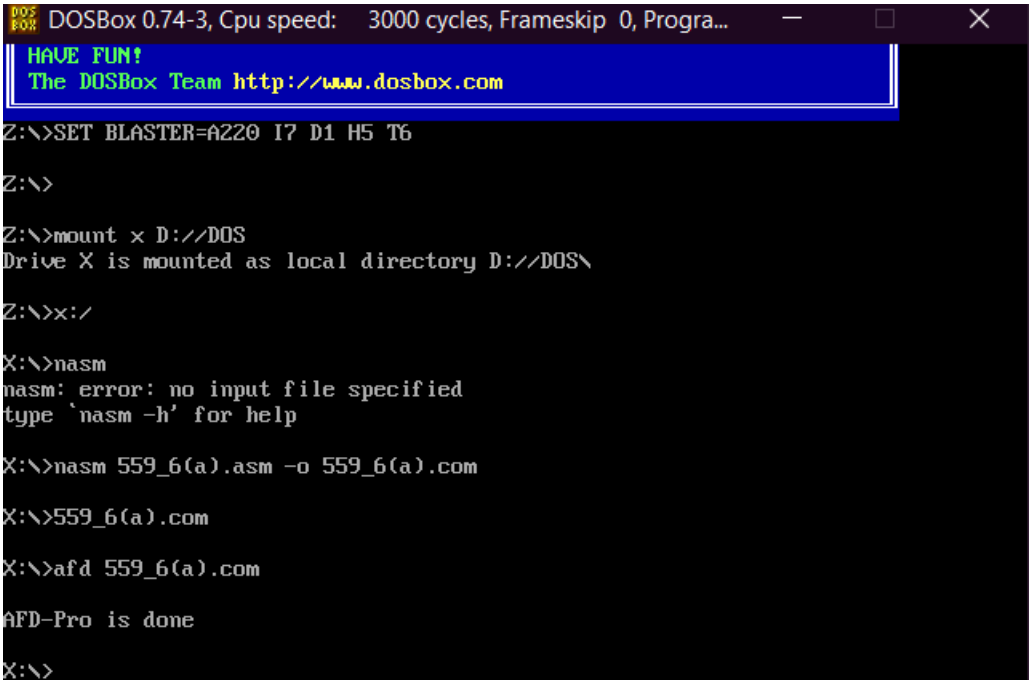
Roll NO – **23p-0559**

SECTION – **BCS(3A)**

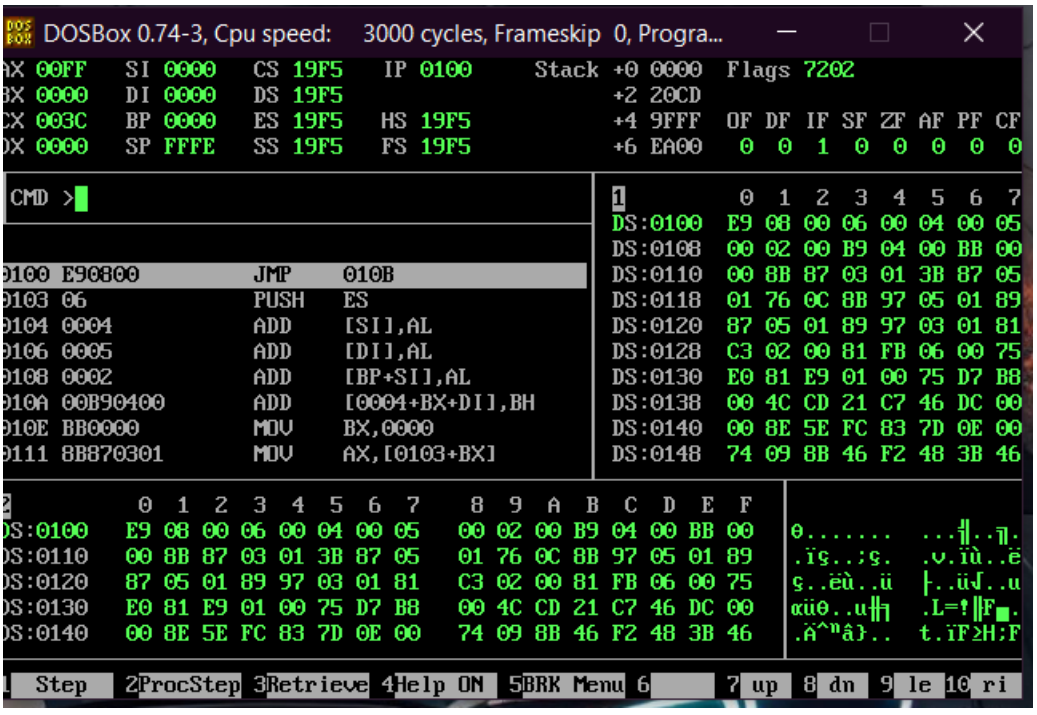
Subject - **Computer Organization and Assembly Language (LAB)**

3.6 (a)

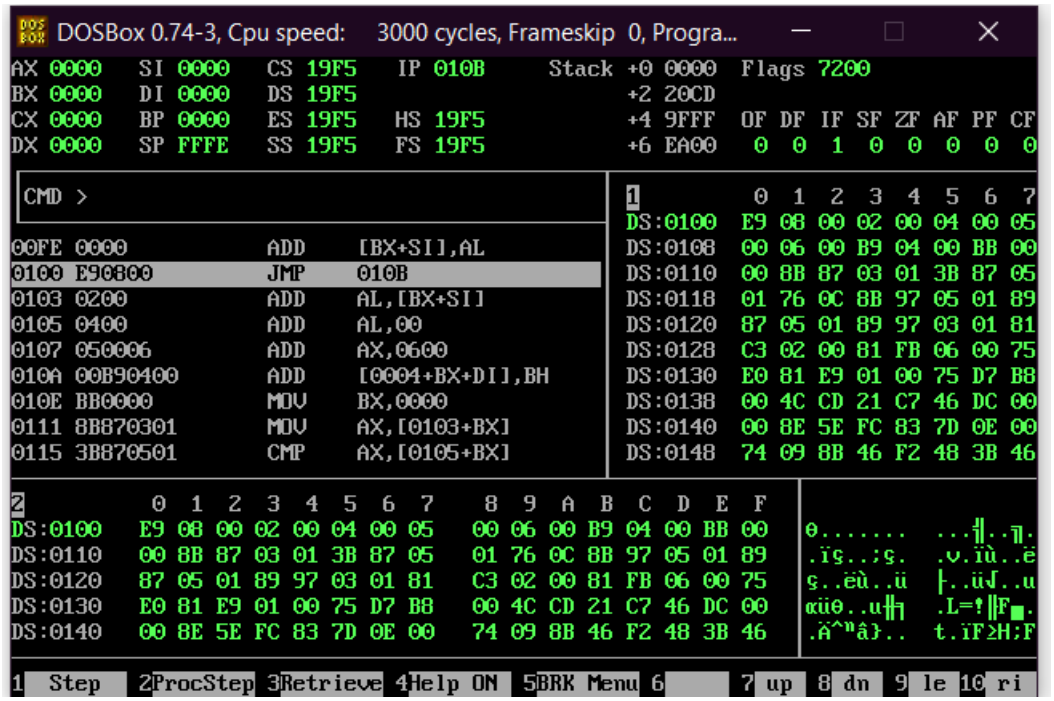
DOS Screenshot:



UnSorted:

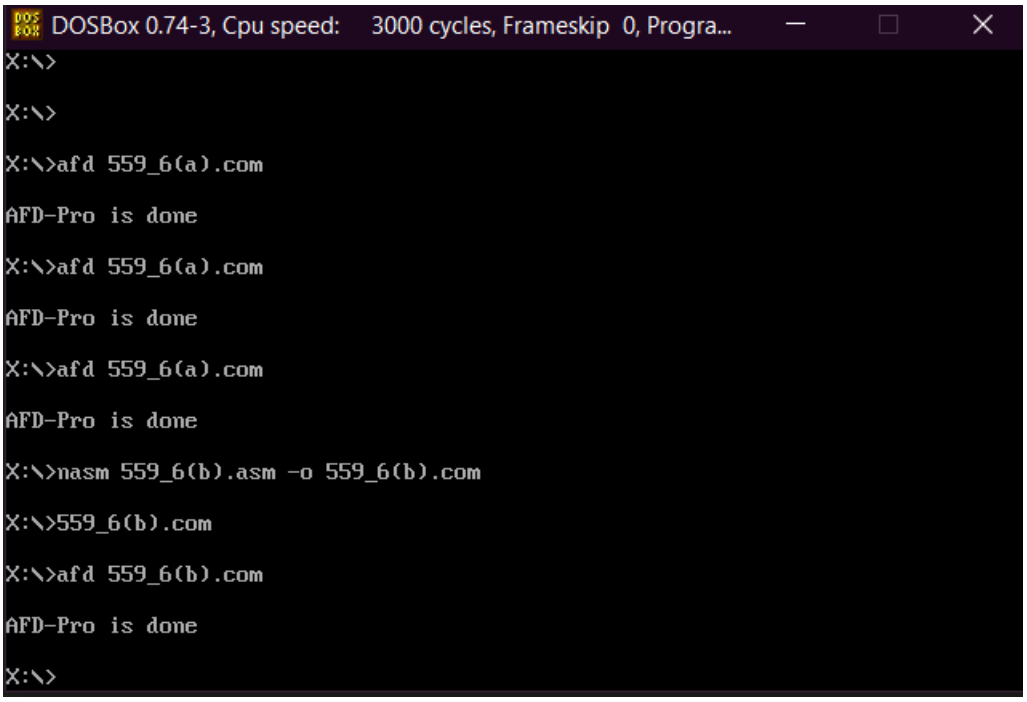


Sorted:

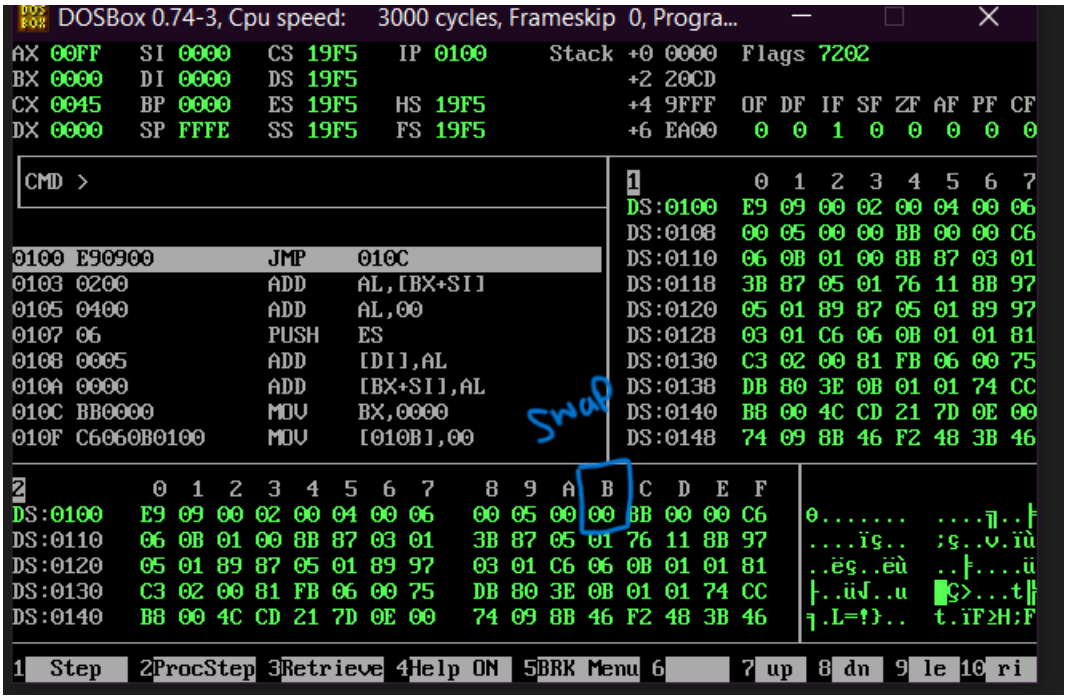


3.6 (b):

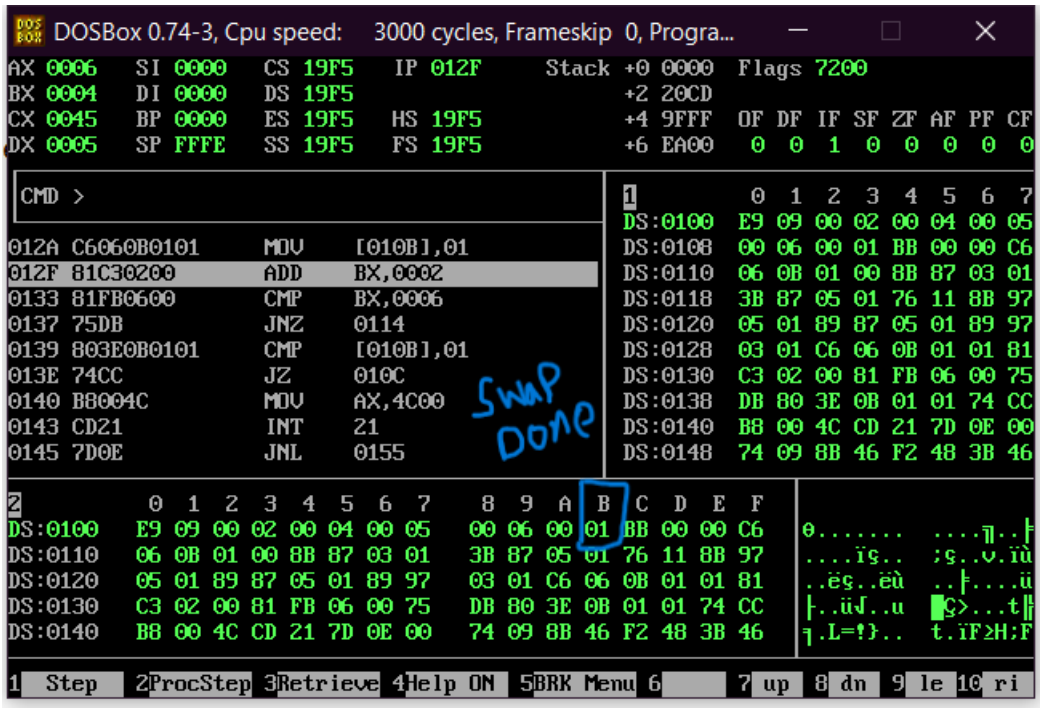
DOS Screenshot:



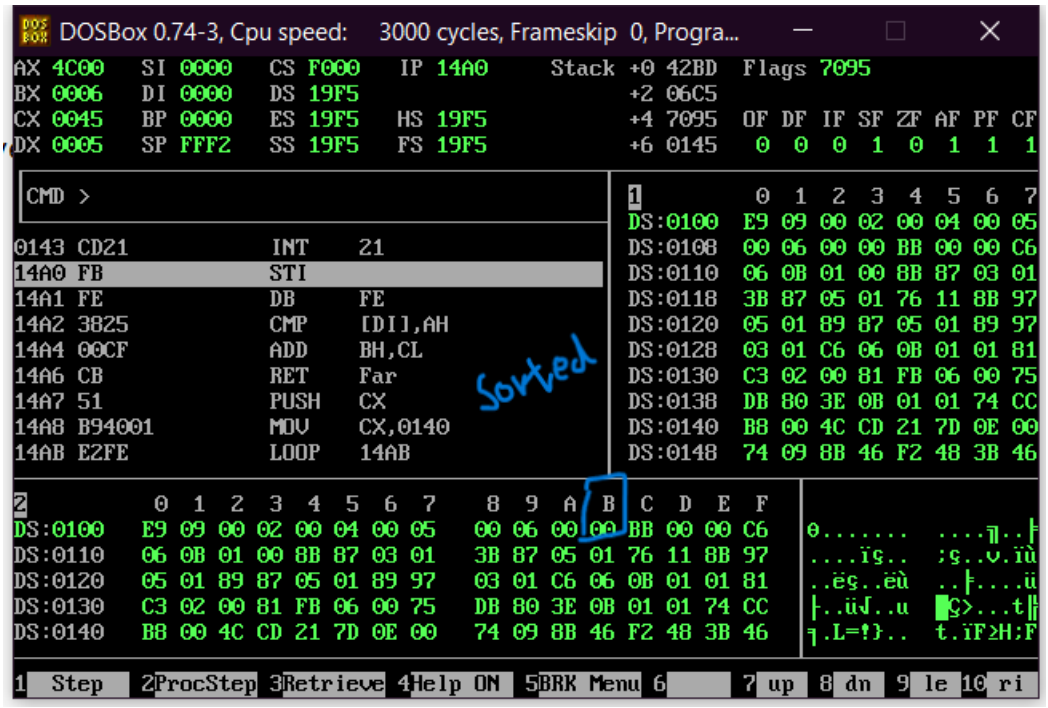
UnSorted:



When SWAP OCCUR



Sorted:



DIFFERNCE:

The first code sorts the array data: dw 6, 4, 5, 2, while the second one sorts data: dw 2, 4, 6, 5. In the first code, the swap flag is not used to track if any swaps happen, whereas in the second code, the swap flag is set to 1 when a swap occurs, helping to determine if another pass is needed. The first code runs a fixed number of passes (4), while the second code checks the swap flag to decide if it should run the outer loop again; if no swaps were made, it stops early. Both codes have comments, but the second code includes more explanations about why certain instructions are used, like the importance of specifying byte and using registers for comparisons. Lastly, the first code includes a commented-out line mov cx, 4, which is not present in the second code, as it doesn't need a fixed number of passes due to the swap flag. In summary, the second code is more efficient because it uses the swap flag to avoid unnecessary passes, making it more adaptable to different data arrangements.