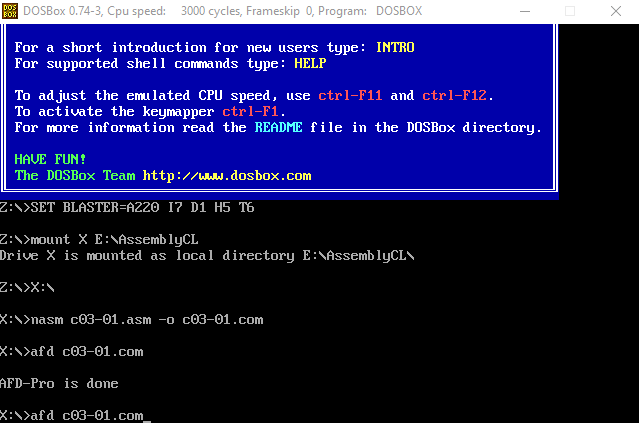
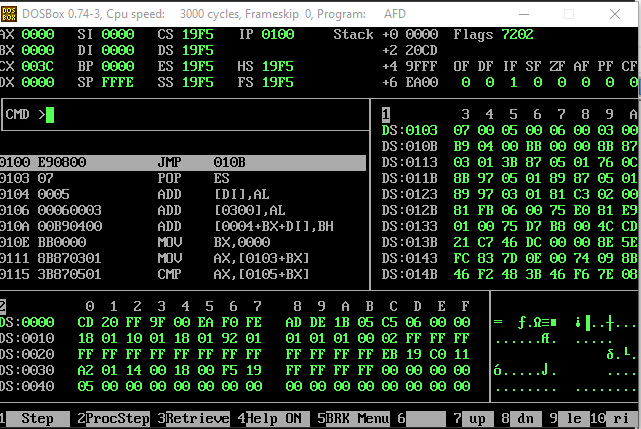
***Lab Task 1: -***

The difference is only that in 3.3 (a) if the values are already in descending order it will still run till the cx has a value of 0 while 3.3 (b) has a different logic in that first of all we are not using the cx and with that we have made a flag other than the inbuilt flags that helps us to refrain from unnecessary passes we initialize the value in the swap flag as 0 and whenever a swap is done its value is made 1 and it will continue in that pass but when there is no swap in the whole inner loop the value will remain 0 in the swap flag , hence at the end while checking if a swap has happened it will continue to another pass but if its value is 0 it will exit as in that state the numbers will be in ascending order.

***(A)***

(Commands)

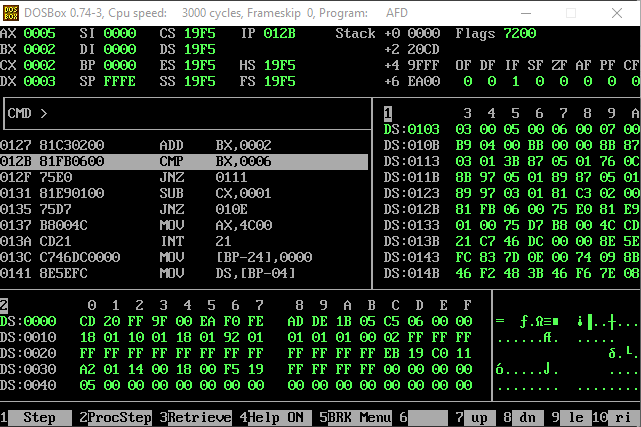


(Initial Data) 

(Pass 1)

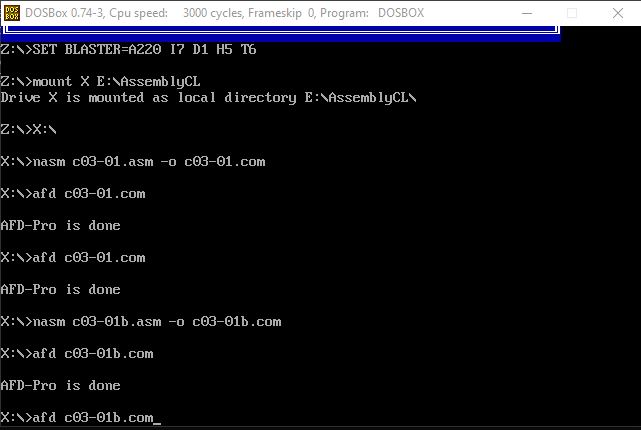


(Pass 2 Till 4rth Pass As The Data Is Arranged But Due To The CX It Will Run Exactly 4 Times)

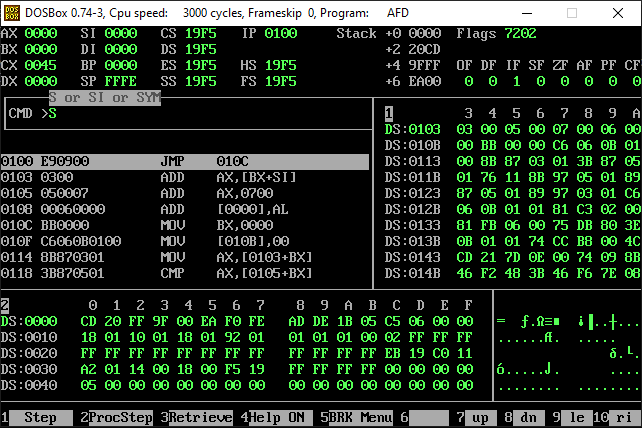


***(B)***

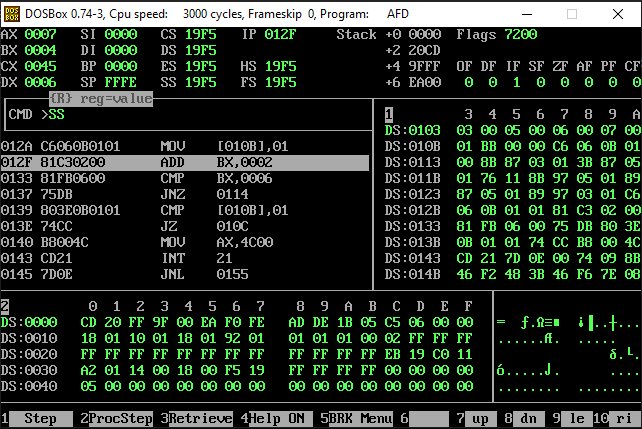
(Commands)



(Initial Data)



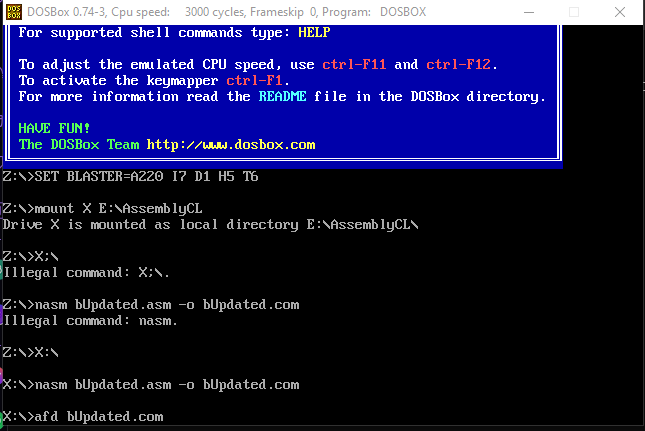
(Pass 1 In A Single Pass The Digits Are Arranged And Enters Into Pass 2 Where the Swap Flag Remains Zero Due To Which The Program Ends)

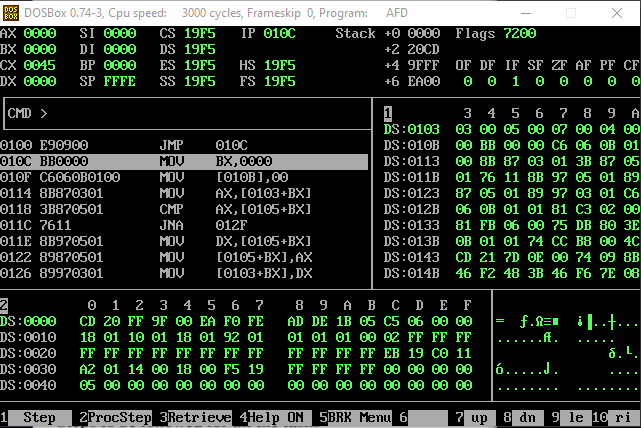


***Lab Task 2: -***

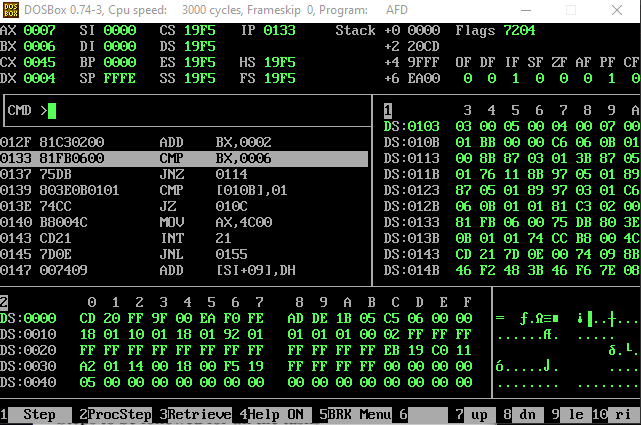
Replacing the jbe with jna will make no difference so we can write jna in place of jbe. With that as the logic is the same as jbe means jump if below or equal and jna means jump if not above which both means the same so both of these will give us the same results.

(Commands)

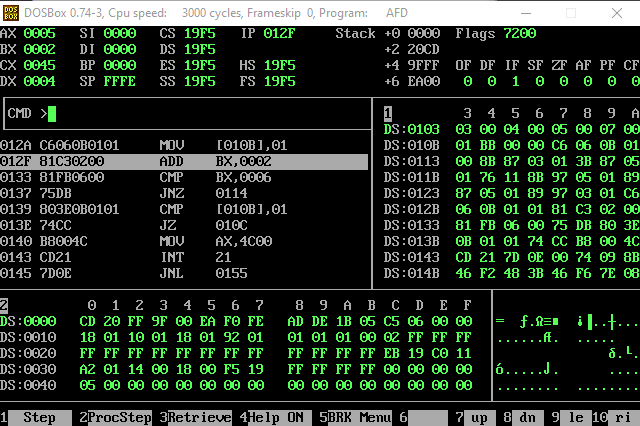


(Initial Data)

(Pass 1 7 takes the place of 4 and 4 takes the place of 7 by swapping)



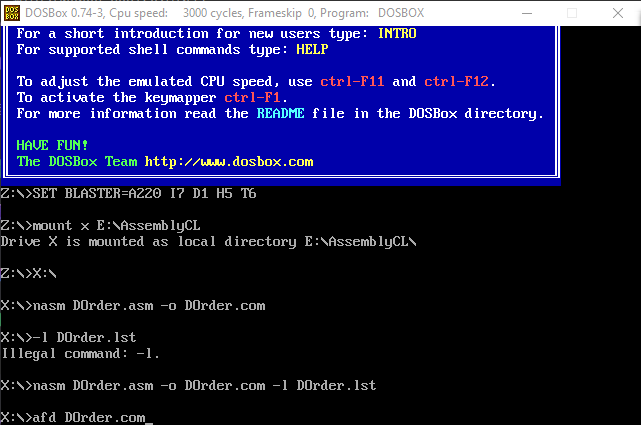
(Pass 3 4 takes the place of 5 and 5 takes the place of 4 by swapping in the next pass the program ends)



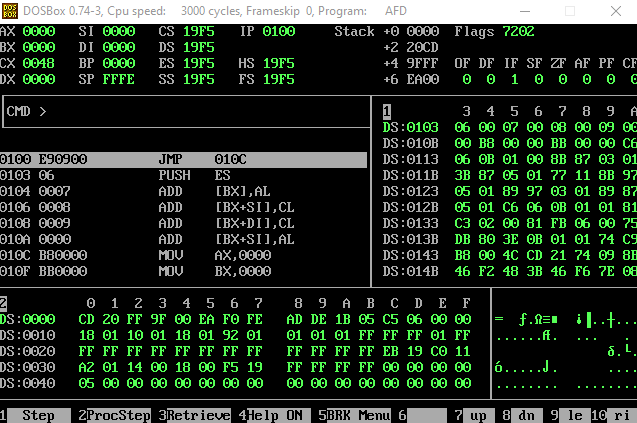
***Lab Task 3: -***

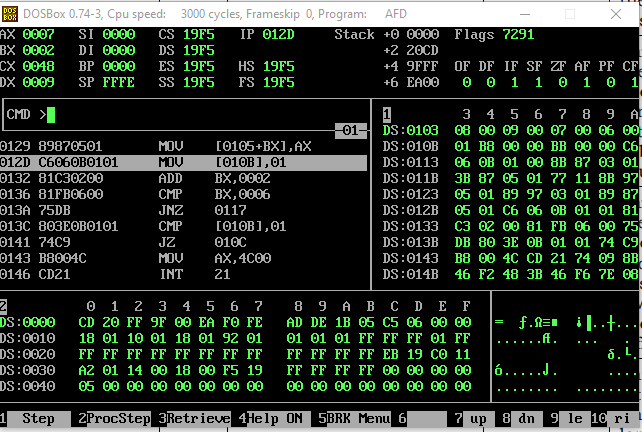
**Descending Order Using JNBE**

(Commands)

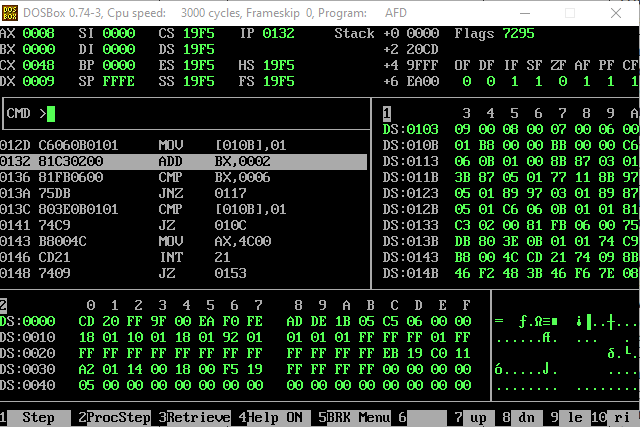


(Initial Data)



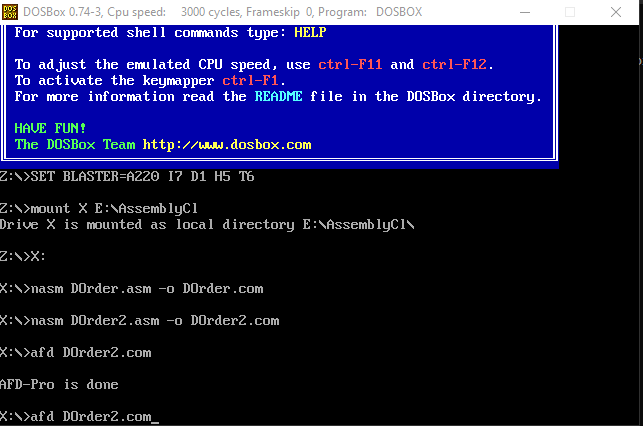
(Pass 1 In The Inner Loop Multiple Swaps Happen That Lead To The Swap Flag Is Raised)

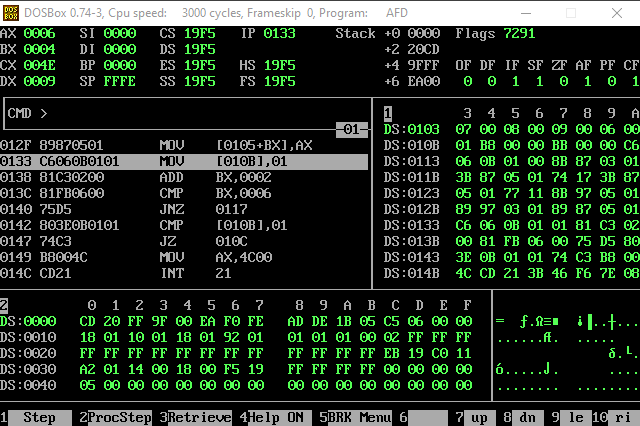
(Pass 2 Same Swapping Occurs And The Swap Flag Is Raised After Which In The Next Pass No Swaps Happen And The Program Exits As The Swap Flag Remains Un Raised Or 0)

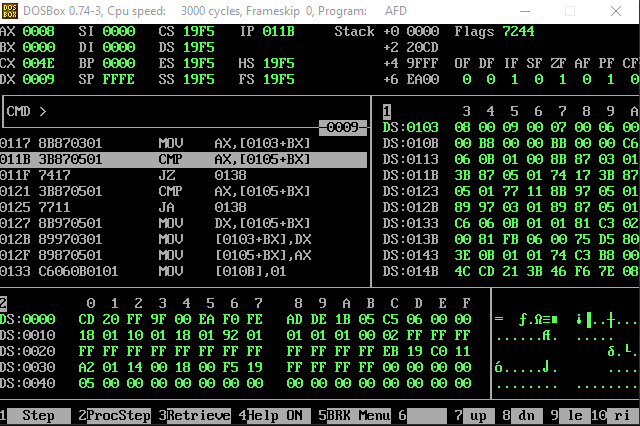


***Lab Task 3: -***

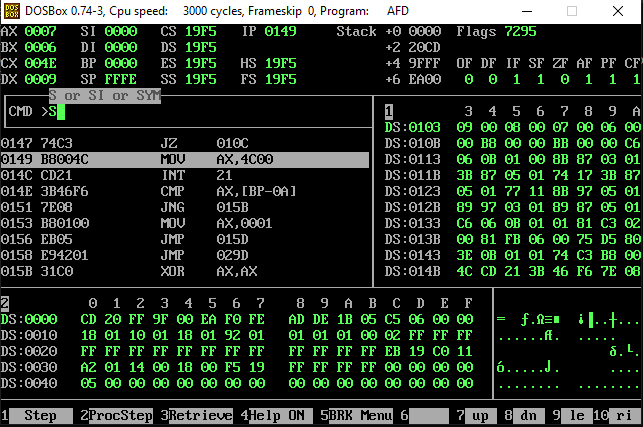
Using 2 jump statements one it will compare and jump if the two numbers are equal to noswap code (je) and the other one is if its greater it will use (ja) to jump if it is greater.

(Commands)

(Initial Data)

(Pass 1 Swap has Happened Swap Flag Was Raised And Pass 2 Has Started Showing The Swap Flag Rest To Zero)

(Pass 2 Caused The Data To Be Completely Arranged In Descending Order And Pass 3 Started With No Swaps That Caused The Swap Flag To Be Remained Unraised After Which The Program Ends)



***-------------------------------------------------------------THE END-------------------------------------------------------------***