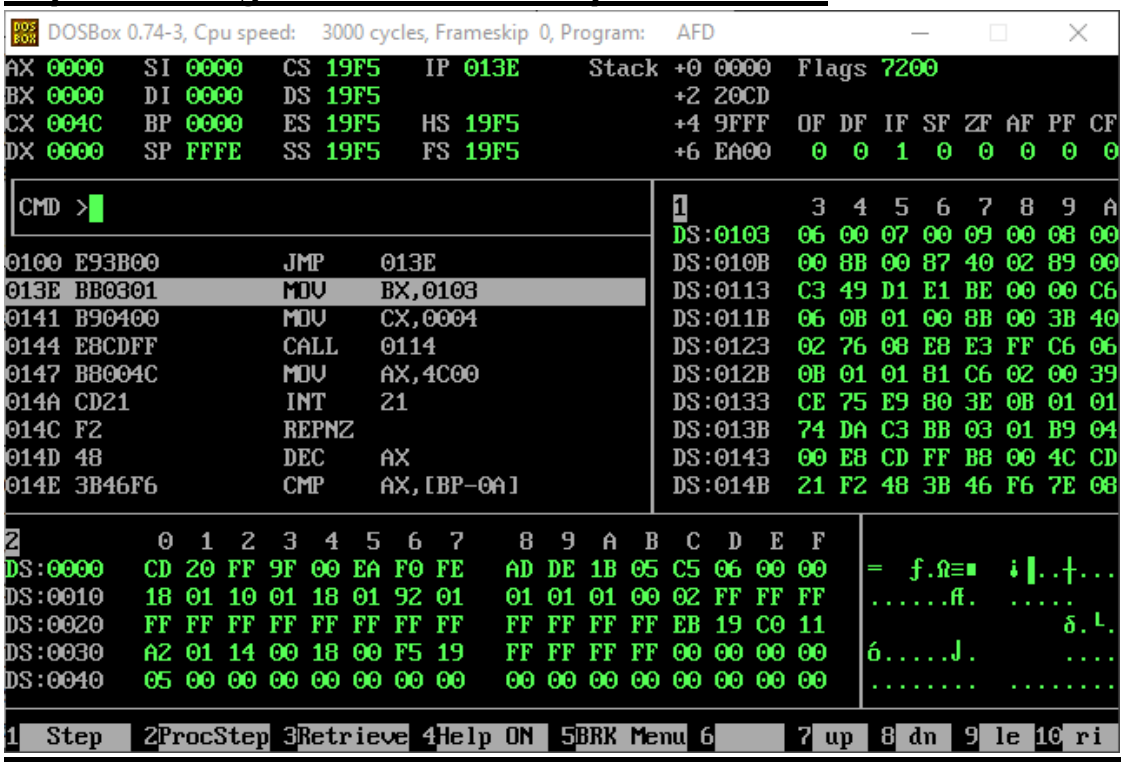
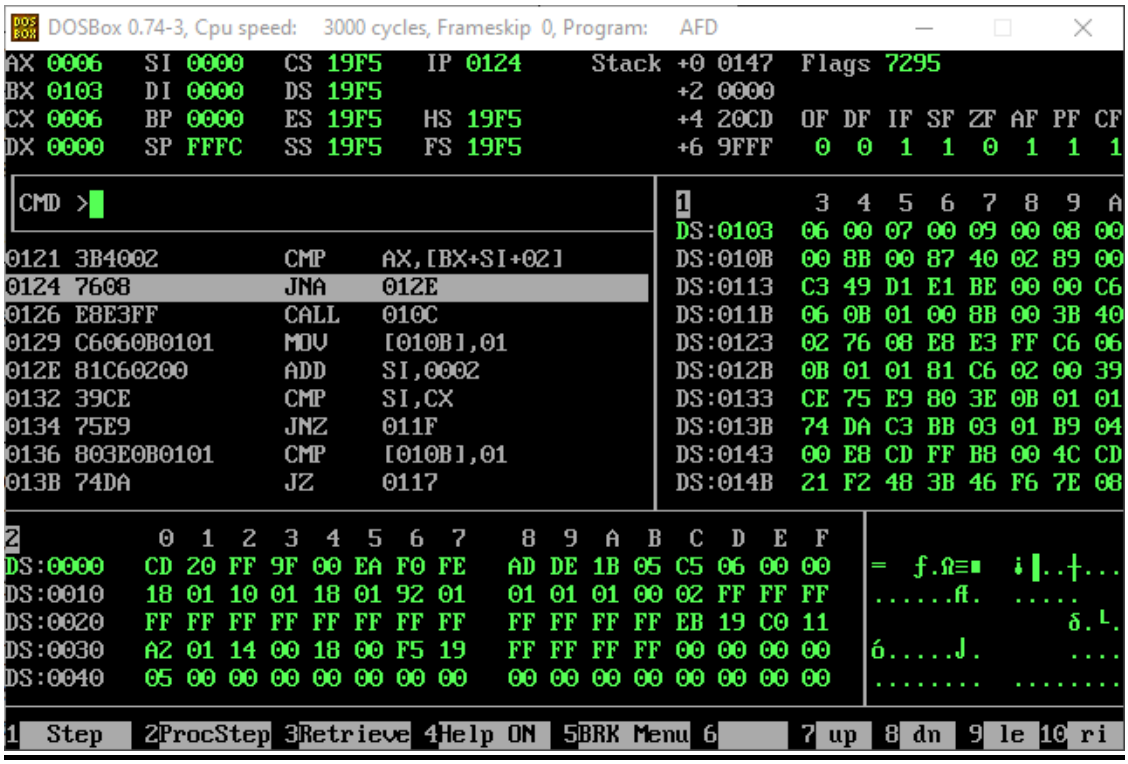
Example 5.3: -

Before Calling The Subroutine Of Bubble Sort

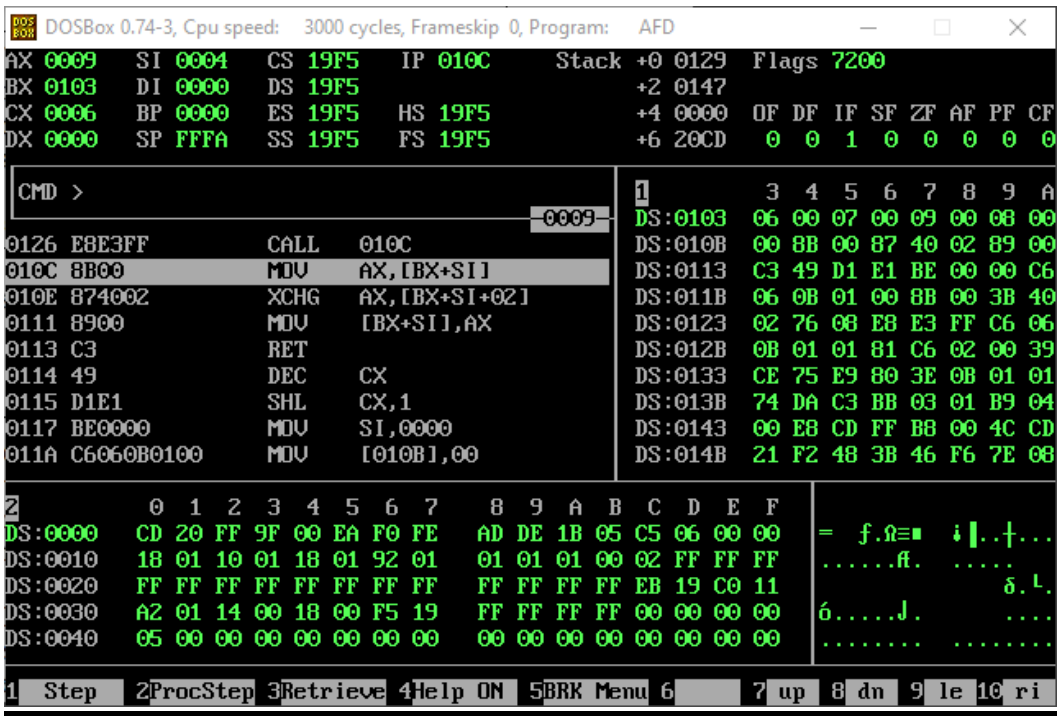


Before Calling The Swap Sub Routine We Are In The bubble Sort Subroutine Notice That The Stack Pointer Has Change

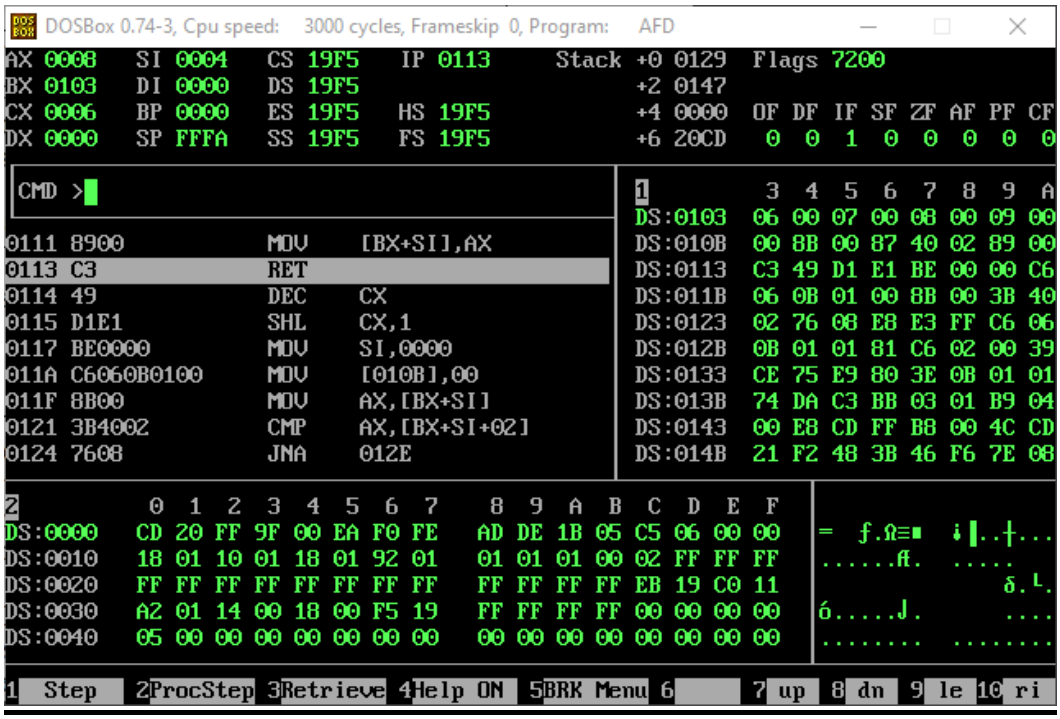


After Calling The Swap Subroutine For The Last Two Values

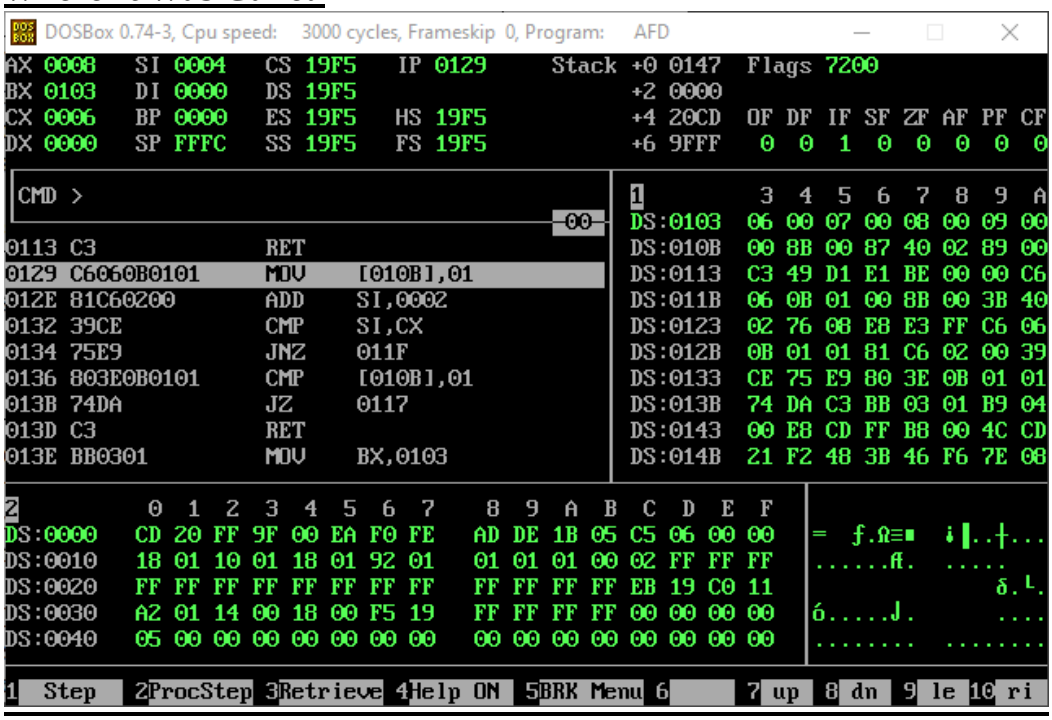
Notice That The Stack Has Changed Once Again For The Swap Subroutine



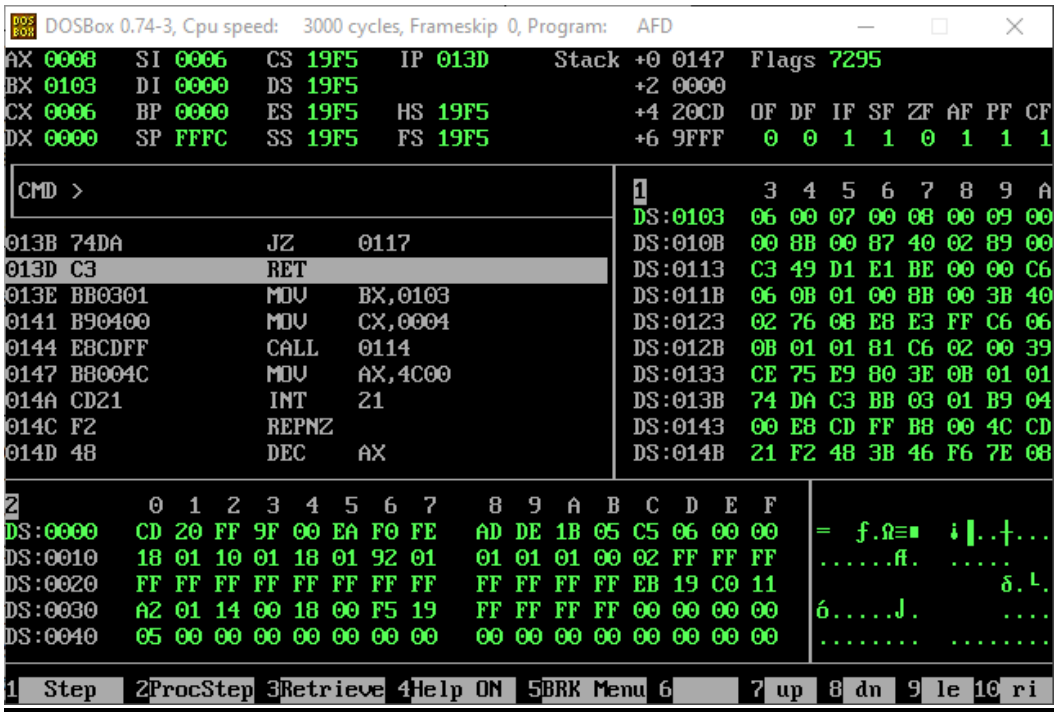
Notice That The xchg command is used to exchange the values between two spaces and after which the value in ax is returned to first space



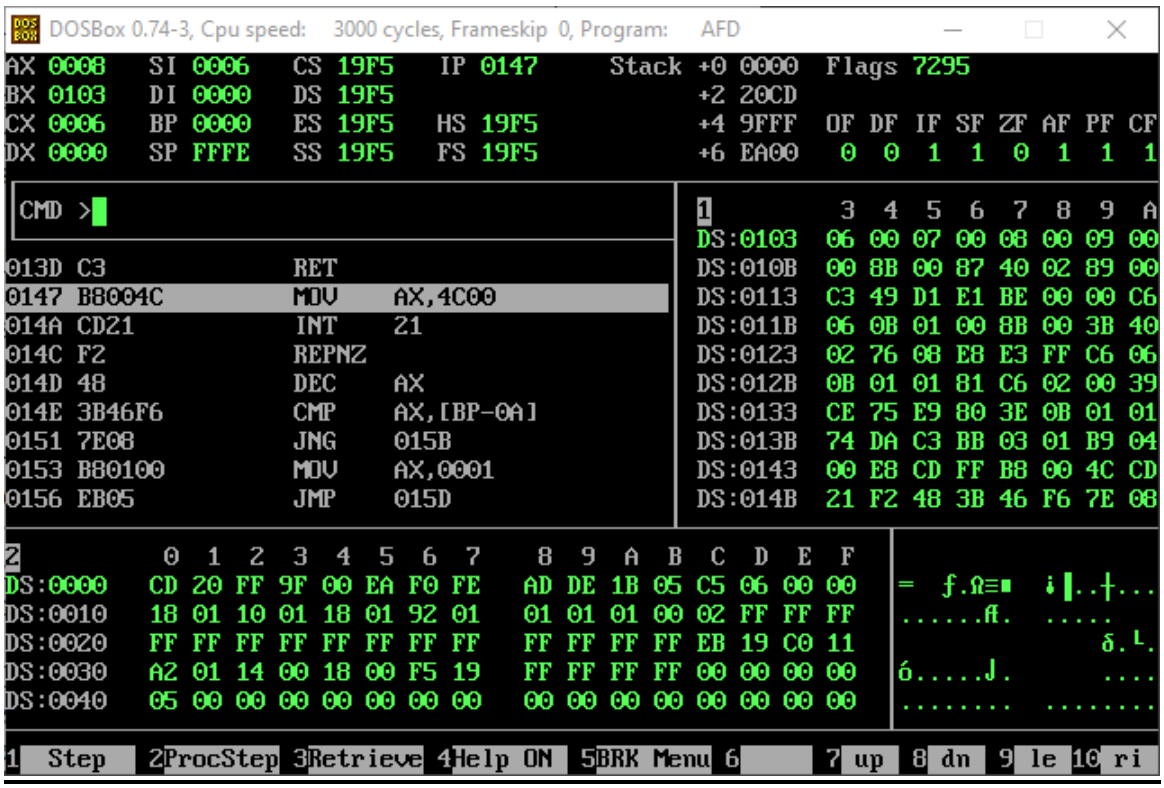
After The Swap Subroutine Has Returned To The Bubble Sort Subroutine Notice That The Stack Pointer Has popped An Address To Come Back From Where It Was Called



As There Were No More Values That Should've Been Swapped So It Didn't Swap Them

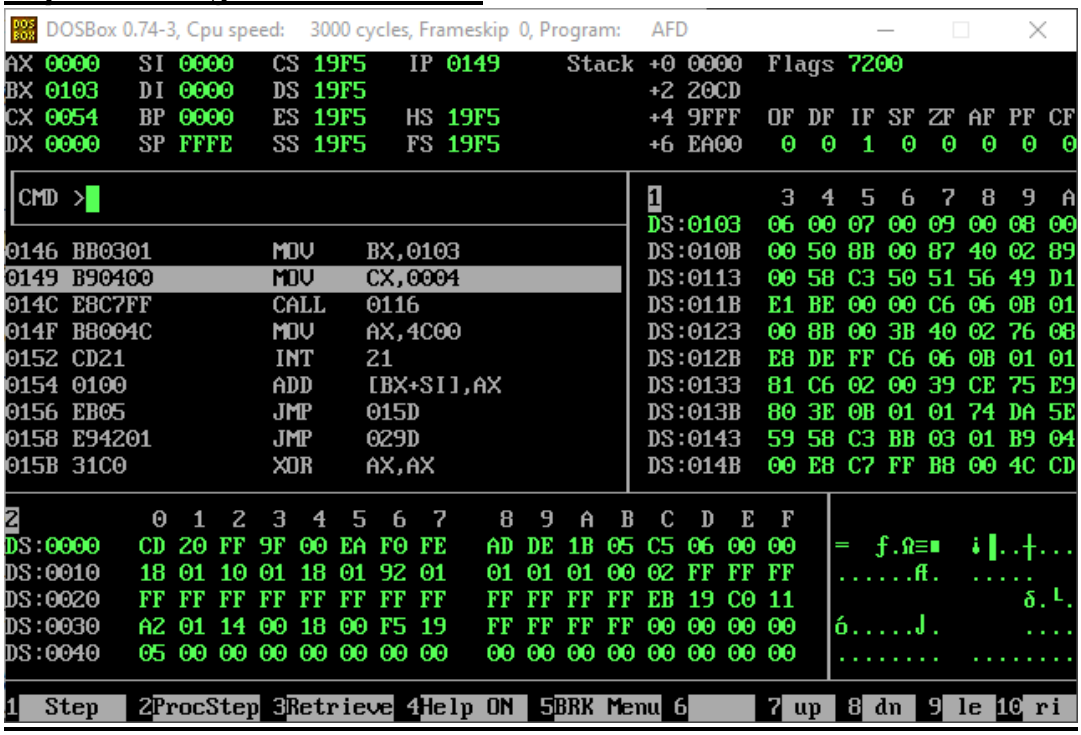


Finally After Returning From The Bubble Sort Subroutine The Stack Has Again Changed So That It Can Return From Where It Was Called

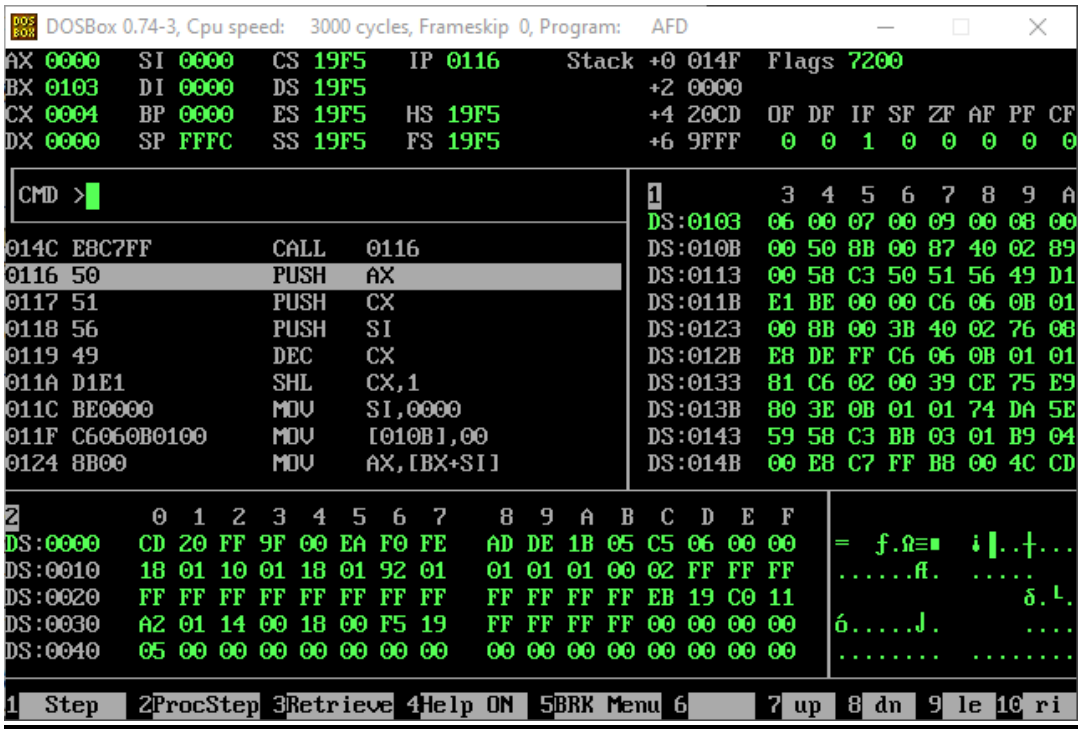


Example 5.4

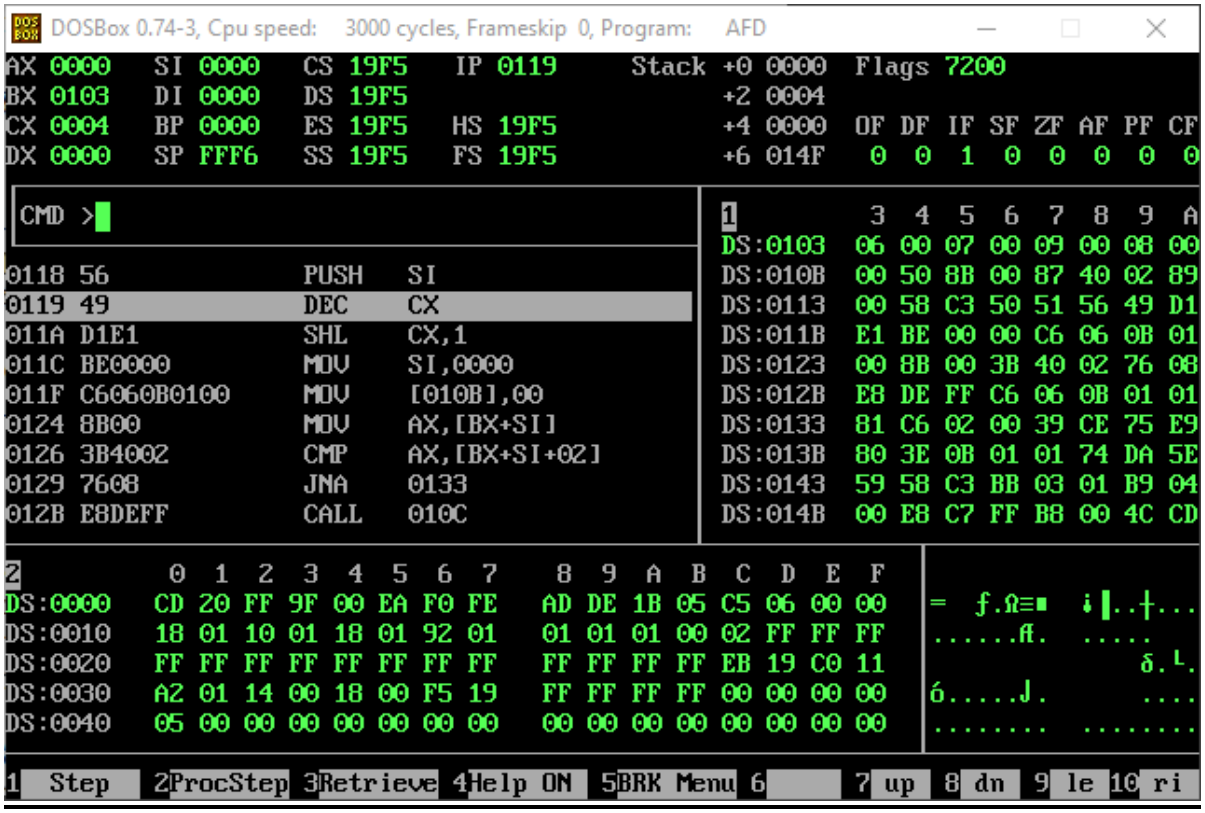
Before Calling The Subroutine



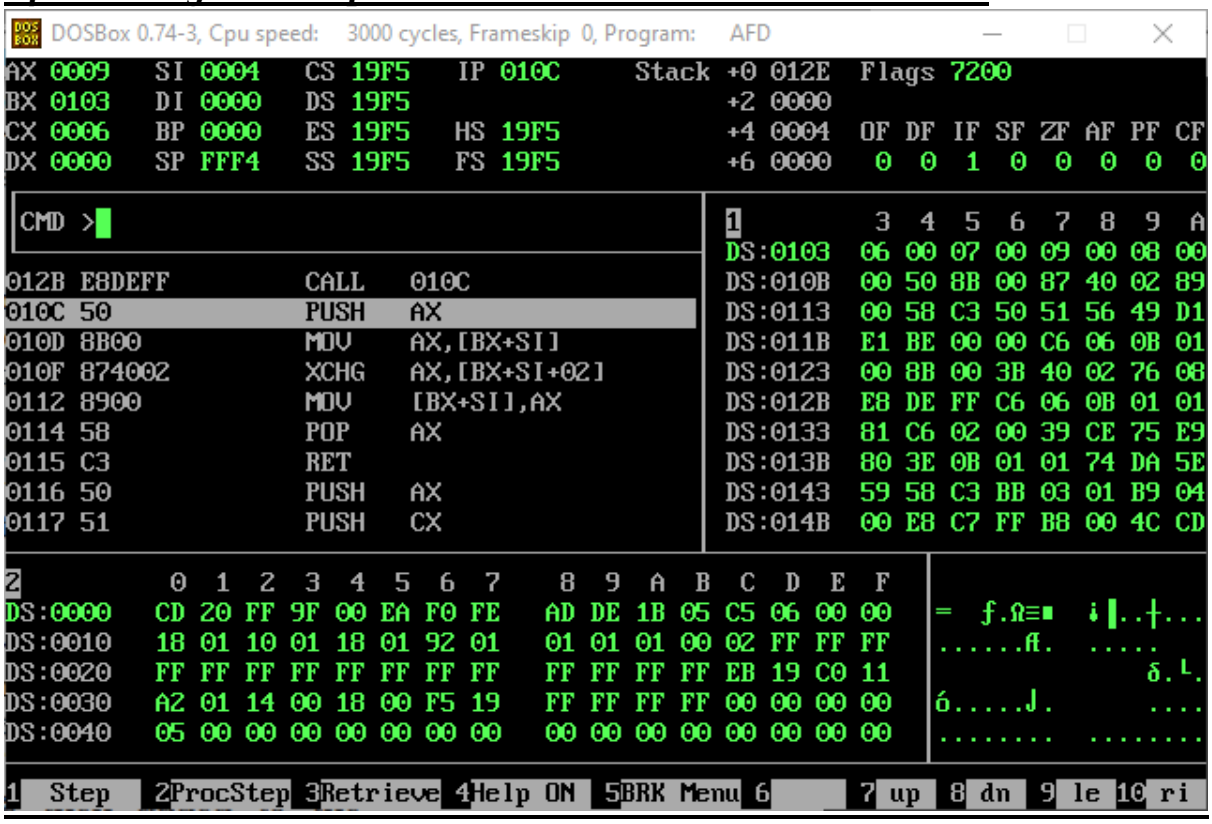
After Calling The Sub Routine Of Bubble Sort



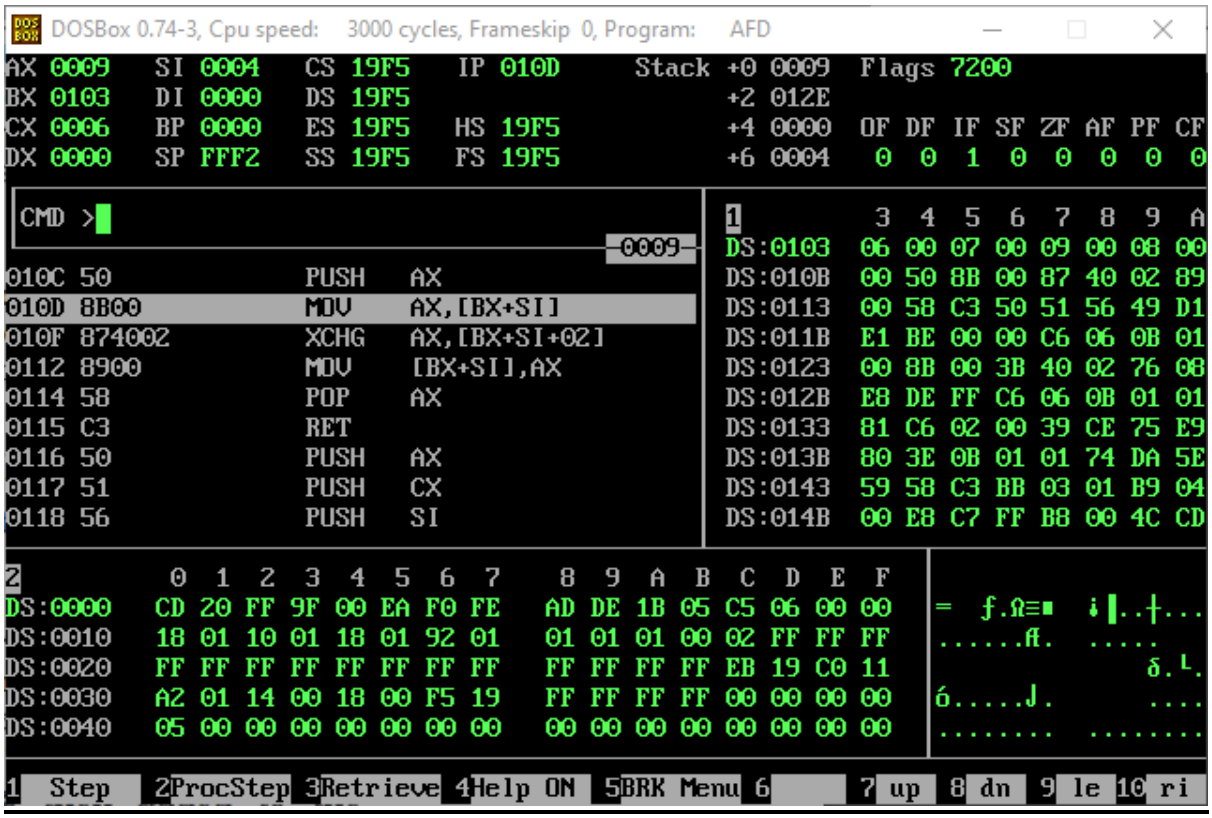
After Pushing Values Of Ax And Cx And Si



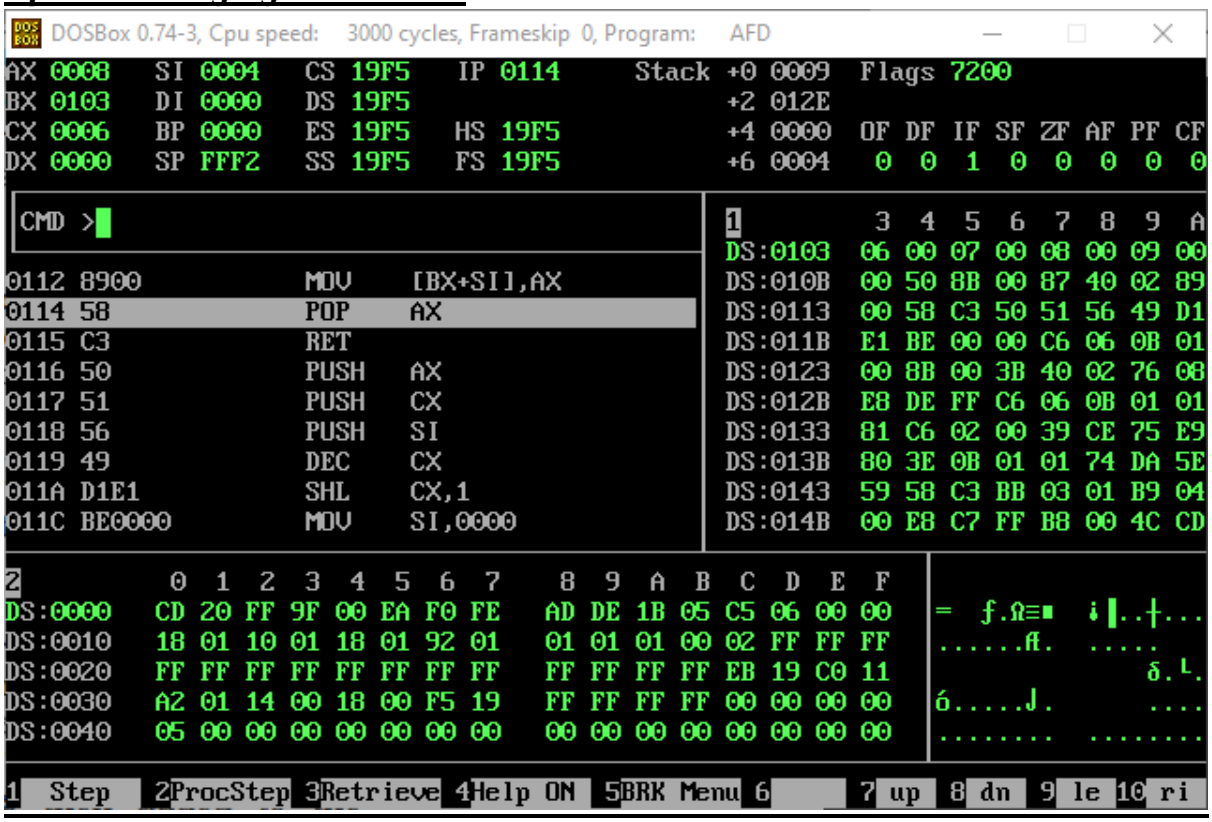
After Calling The Swap Subroutine For The Last Two Numbers



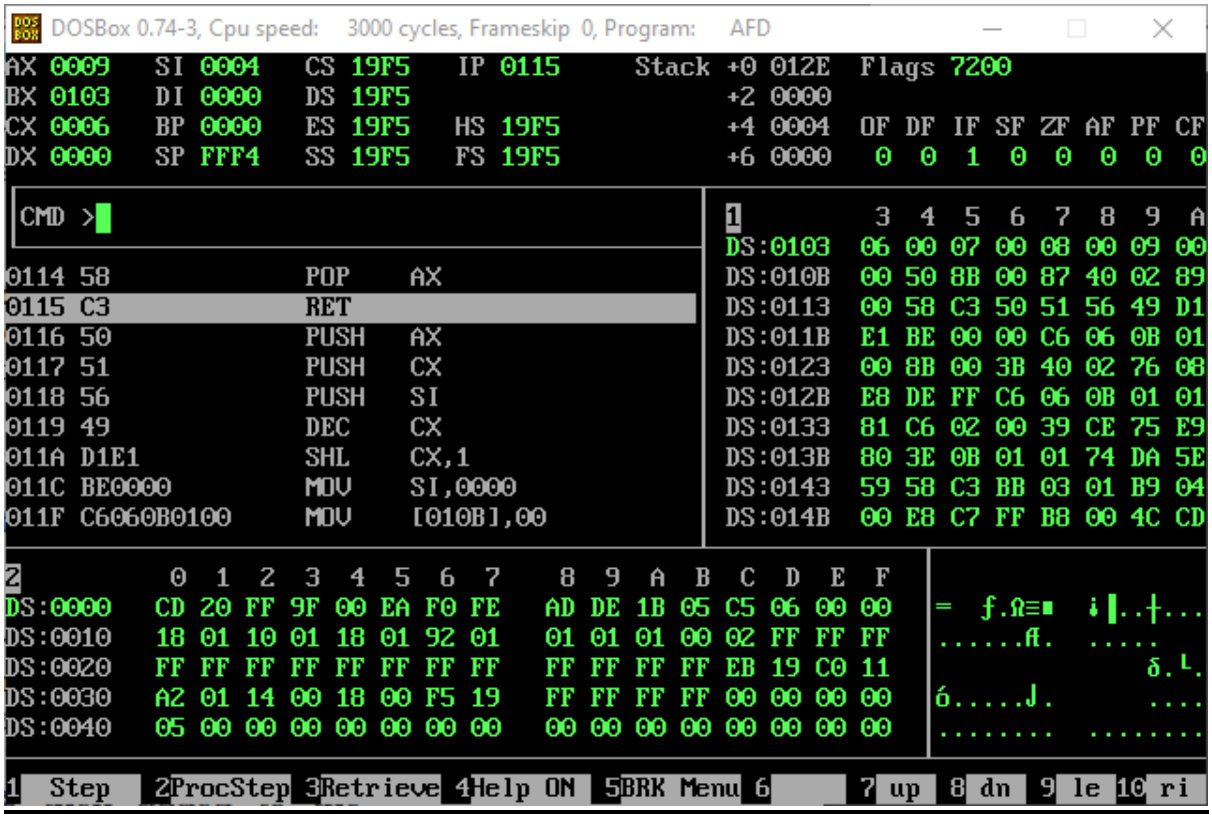
Again After Pushing The Value Of Ax



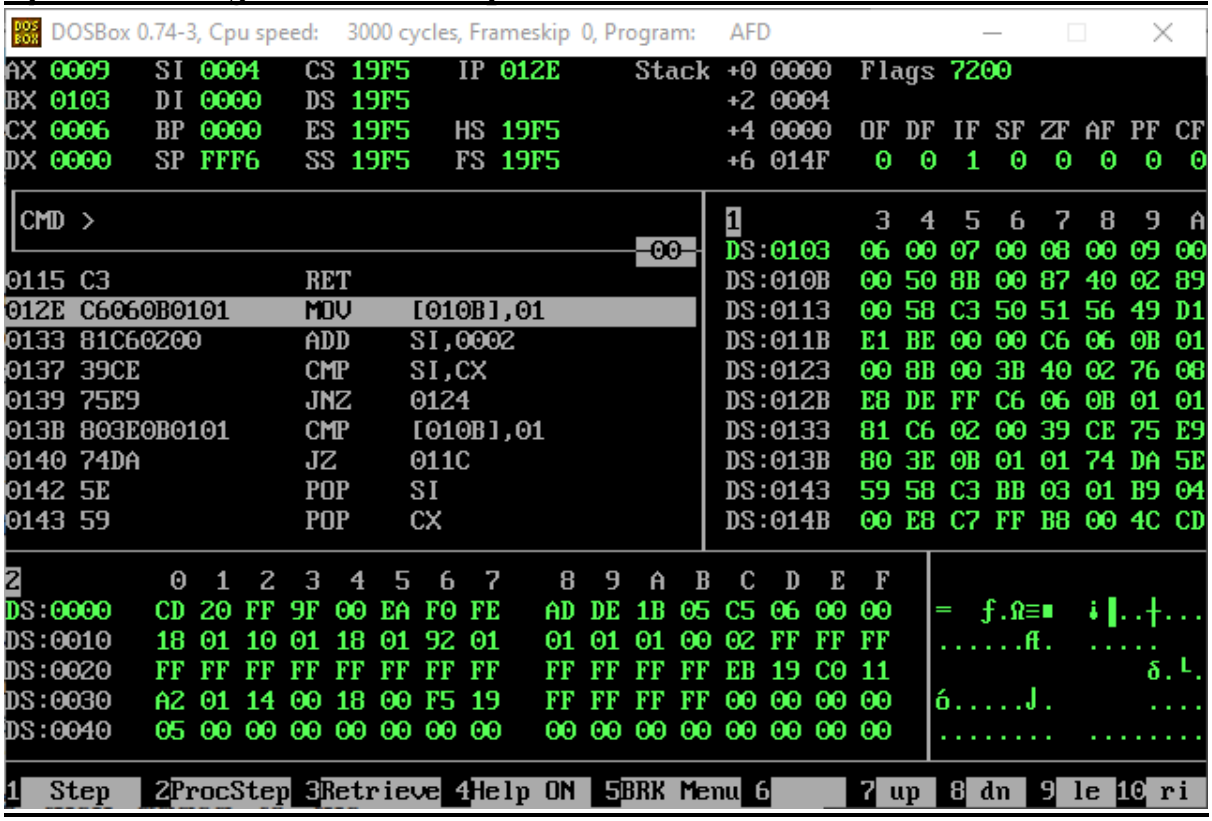
After Exchanging The Values



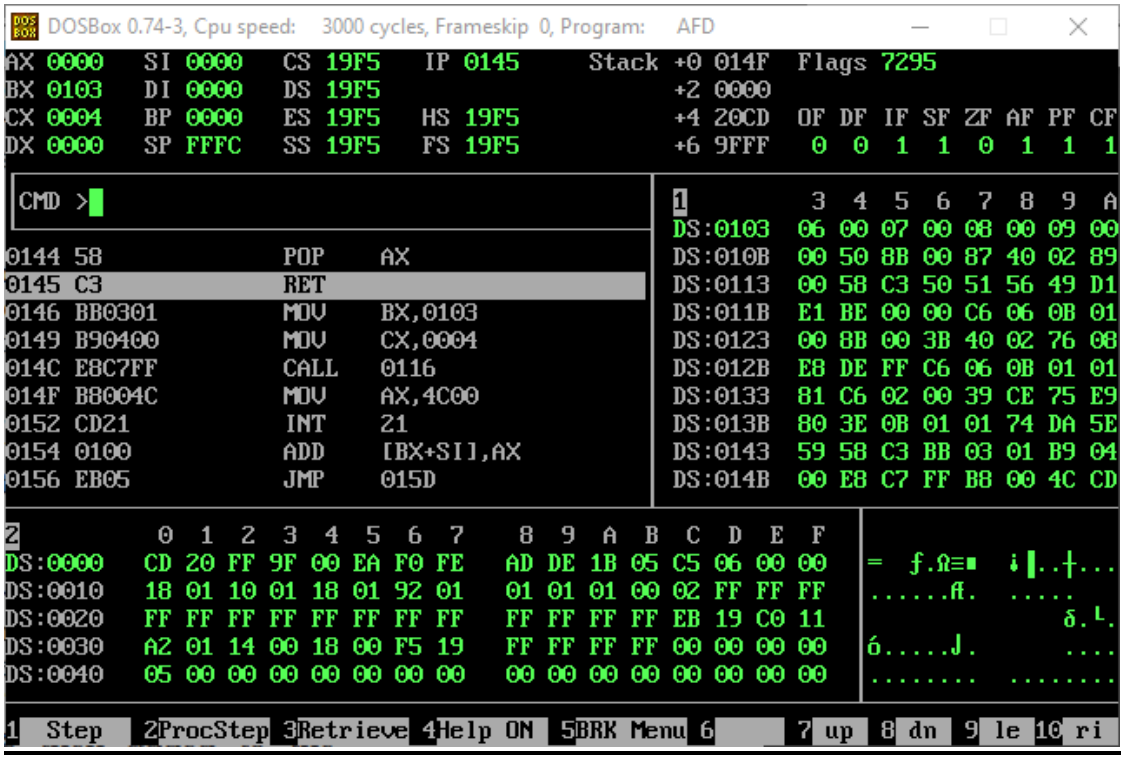
Popping The Value Of Ax In The Swap Subroutine As The Value In Ax Before The Subroutine Is Again Put In Ax



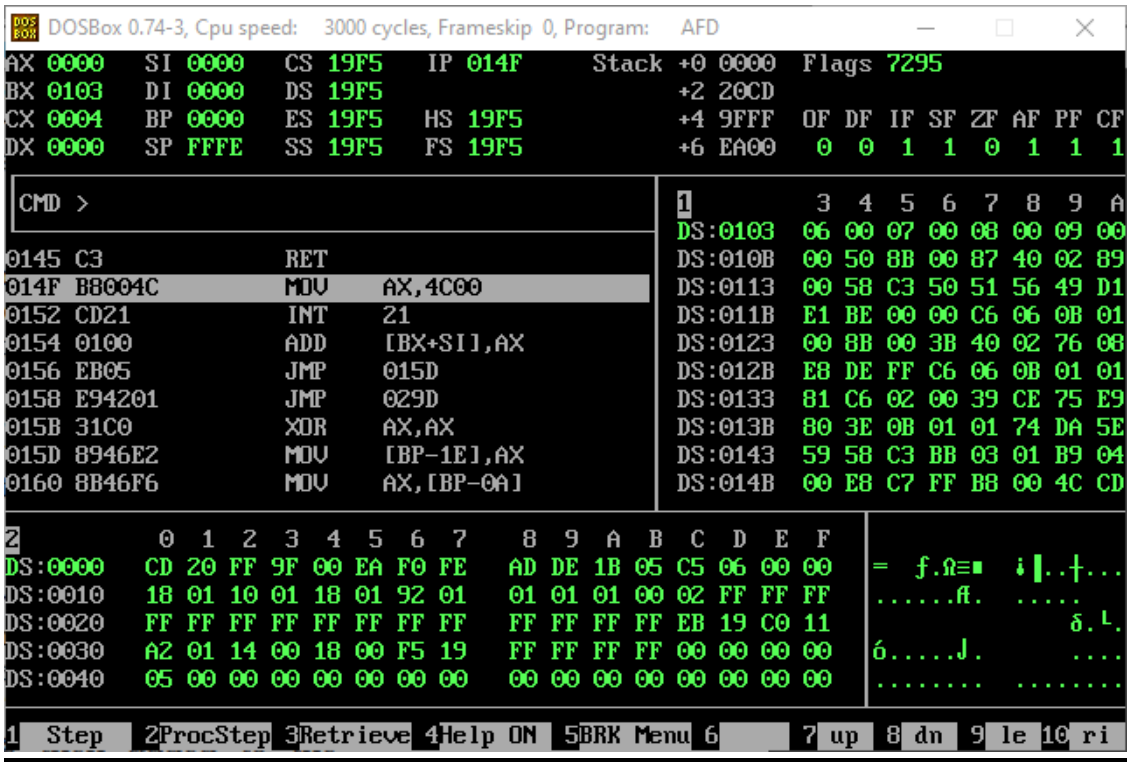
After Returning From The Swap Subroutine Into The Bubble Sort Subroutine



In The Bubble Sort Subroutine As The Data Is All In Ascending Order No Swaps Other Then The One Mentioned Above is Performed And The Values Of SI CX and AX Are Popped As They Were Previously Pushed Into The Stack To Maintain The Before Values Of These Registers



After Returning From The Bubble Sort Subroutine The Stack Has Changed Again As to Return To The Place From Where It Was Called



Note: -

The main difference in both the examples is that in the first example we are not using stack to store the previous data in the registers while in the second example we are using the stack to store the values of the register in the stack and after using these registers for our required functions we pop back the values of the registers from the stack back to the registers to maintain the values in the registers so they are not overwrote and it stays as they were before we used them.