Nirma University
Institute of Technology
Department of Electronics and Communication Engineering

2EC404 - Microprocessors & Microcontrollers

Sessional Assignment

Name: Vincent Abraham

Roll no.: 18BEC122

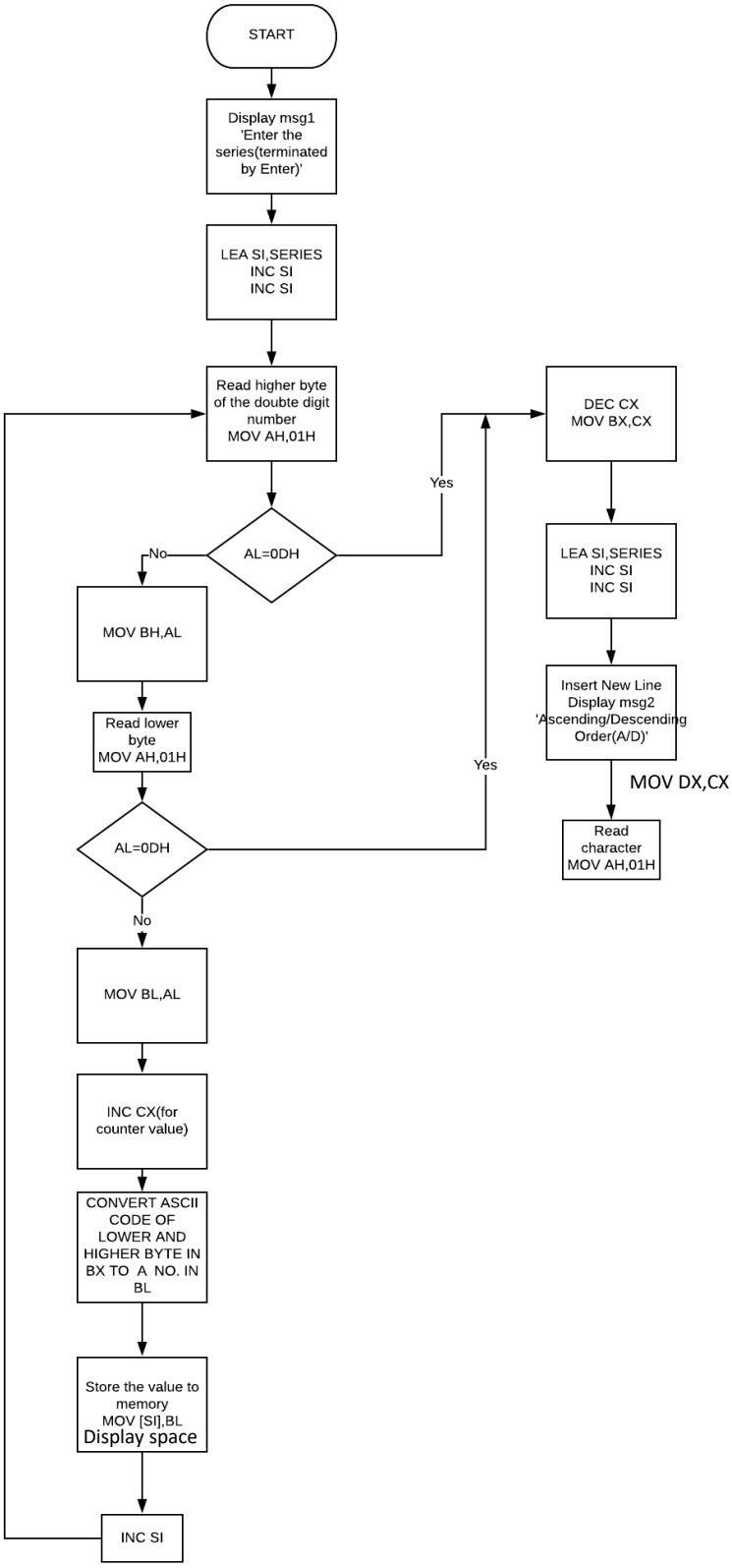
Question:

To develop an interactive program in 8086 assembly language for arranging a given series in ascending and descending order as follows:

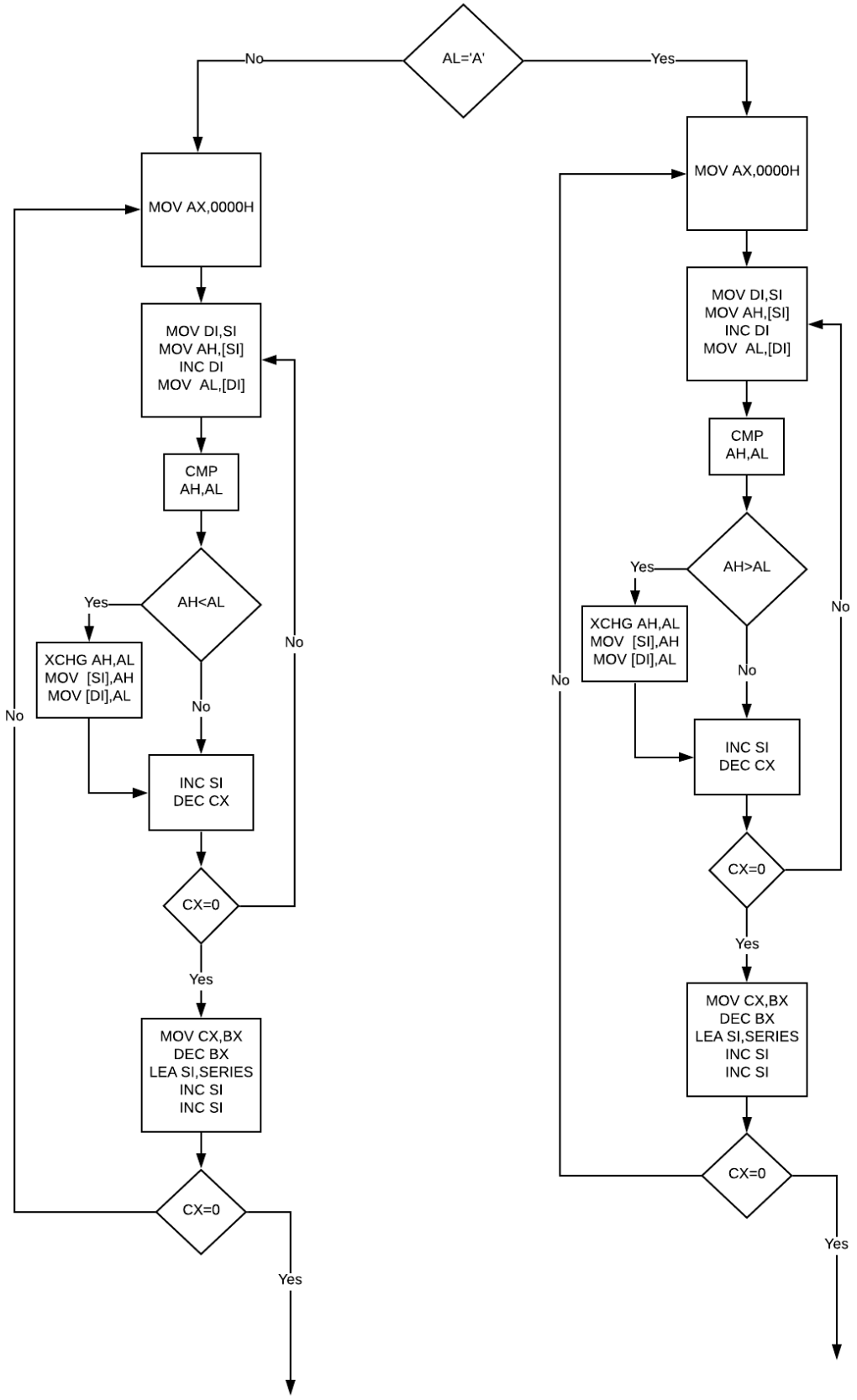
- Enter the series (terminated by enter):
- Ascending/descending order (A/D):
- The ordered series is:
- Do you want to continue (y/n)?
- Continue if yes and exit if no.

Flowchart:

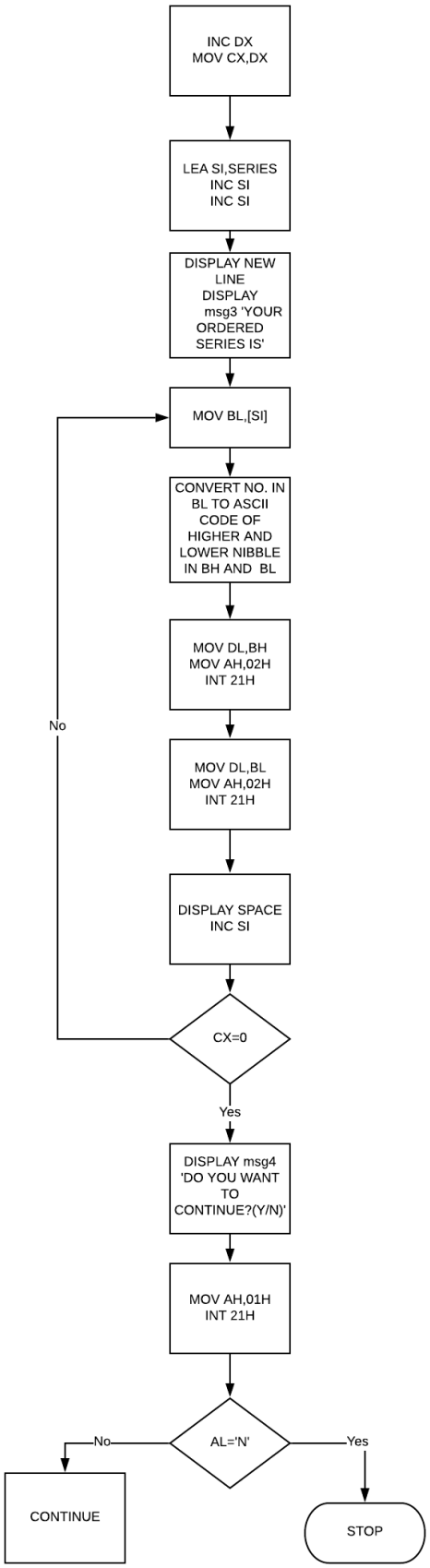
❖ The first part of the program algorithm will be scanning and assembling the characters entered by the user.



❖ The second part will be sorting the series in ascending or descending order.



❖ The final part of the program will be displaying the ordered series and asking if the user wants to continue or not.



Program in words:

- This program is for scanning a double-digit HEX or BCD number entered by the user and sorting it in ascending or descending order as entered by the user. All the numbers should be entered in double digit form, even if it's a single character, it should be entered as 01, 05, 0A, ...etc.
- For storing the series, a buffer is created in the data segment of maximum length of 50 bytes and 45 spaces are allotted with random values for entering the characters. In the data segment all the messages which have to be displayed are also stored.
- First, the message 'Enter the series (terminated by Enter):' is displayed and then the scanning portion of the program begins. For scanning a double-digit number, the function MOV AH,01H INT 21H is repeated twice for the higher and the lower nibble. These values are stored into a separate BX register from AL where they are converted into BCD or HEX from their respective ASCII values and then assembled into a single byte which is stored in register BL. After this, the number in BL is stored into the series buffer, a space is displayed on the screen and this whole process is repeated until the Enter key is pressed. When the Enter key is pressed it goes to the next part of the program.
- Now, the message 'Ascending/Descending Order(A/D)' is displayed and depending upon the character A or D entered by the user, it goes to the loop for ascending or descending order.
- The method used for sorting here is bubble sort. To explain this, if we are sorting for ascending order then, the no. at the 1^{st} position pointed by SI is compared with the no. at the 2^{nd} position pointed by DI, and if $1^{st} > 2^{nd}$ ($1^{st} < 2^{nd}$ for sorting in descending order) the numbers are swapped. After this, SI and DI are incremented and this procedure is repeated N-1 times for all the positions and at the end we can find that the no. at the last position is the highest. So, here the original counter value is decremented by 1 and the whole procedure is repeated N-2 times. This whole process is repeated until the counter value becomes zero.
- When the sorting is completed, we move to the final part of the program where first, the message 'Your ordered series is:' is displayed and after that we take the numbers from the memory, and store it BL register. After that, the higher and lower byte of the BCD or HEX no. is converted to its respective ASCII values which is stored in BH, BL which is then moved to register DL for displaying each byte. Here MOV AH,02H INT 21H is done twice for the double-digit no and after the two digits are displayed, a space is inserted.
- After the sorted series is displayed, the message 'Do you want to continue(Y/N)?' is displayed. If the user enters 'Y' then the program is run again from the start and if 'N' is pressed, the program is terminated using MOV AH,4CH INT 21H.

Code:

```
Data Segment
        msg1 db 'Enter the series(terminated by Enter): $'
        msg2 db 'Ascending/Descending Order(A/D): $'
        msg3 db 'Your ordered series is: $'
        msg4 db 'Do you want to continue?(Y/N): $'

        series db 50,?,45 DUP(?)
Data Ends

Code Segment
        assume cs:code,ds:data,es:data
        start: mov ax,data
               mov ds,ax
               mov es,ax
          cont:mov cx,0000h

               mov dl,0ah
               mov ah,02h
               int 21h
               lea dx,msg1
               mov ah,09h
               int 21h
               mov dx,0000h

               lea si,series
               inc si
               inc si
               scan:mov ah,01h
                    int 21h
```

```
        cmp al,0dh
        jz over
        mov bh,al
        mov ah,01h
        int 21h
        cmp al,0dh
        jz over
        mov bl,al
        inc cx
        cmp bl,40h
        JA l1
        and bl,0fh
        jmp end1
    l1:and bl,0fh
        add bl,09h
   end1:cmp bh,40h
         JA l2
         and bh,0Fh
         jmp end2
         l2:and bh,0Fh
            add bh,09h
         end2:shl bl,1
              shl bl,1
              shl bl,1
              shl bl,1
              shr bx,1
              shr bx,1
              shr bx,1
              shr bx,1
              mov [si],bl
              inc si
```

```
                mov dl,20h
                mov ah,02h
                int 21h
                jmp scan

 over:dec cx
      mov bx,cx
      lea si,series
      inc si
      inc si
      mov dl,0ah
      mov ah,02h
      int 21h
      lea dx,msg2
      mov ah,09h
      int 21h
      mov dx,cx
      mov ah,01h
      int 21h
      cmp al,'A'
      JZ asc
      cmp al,'D'
      JZ dsc

back:mov ax,0000h
     asc:mov di,si
     mov ah,[si]
     inc di
     mov al,[di]
     cmp ah,al
     ja swap
```

```
        jmp exit
        swap:XCHG ah,al
             mov [si],ah
             mov [di],al
        exit:inc si
             loop asc
        mov cx,bx
        dec bx
        lea si,series
        inc si
        inc si
        loop back
        jmp display1

back1:mov ax,0000h
      dsc:mov di,si
      mov ah,[si]
      inc di
      mov al,[di]
      cmp ah,al
      jb swap1
      jmp exit1
      swap1:XCHG ah,al
            mov [si],ah
            mov [di],al
      exit1:inc si
            loop dsc
      mov cx,bx
      dec bx
      lea si,series
      inc si
```

```
        inc si
        loop back1

display1: inc dx
          mov cx,dx
          lea si,series
          inc si
          inc si
          mov dl,0ah
          mov ah,02h
          int 21h
          lea dx,msg3
          mov ah,09h
          int 21h

display:mov bl,[si]
        mov bh,bl
        and bl,0Fh
        shr bh,1
        shr bh,1
        shr bh,1
        shr bh,1
        cmp bl,09h
        JA hex1
        or bl,30h
        jmp done1
        hex1:sub bl,09h
             or bl,40h
        done1:cmp bh,09h
             JA hex2
             or bh,30h
```

```
        jmp done
        hex2:sub bh,09h
             or bh,40h
done:mov dl,bh
     mov ah,02h
     int 21h
     mov dl,bl
     mov ah,02h
     int 21h
     mov dl,20h
     mov ah,02h
     int 21h
     inc si
     loop display

   mov dl,0ah
   mov ah,02h
   int 21h
   lea dx,msg4
   mov ah,09h
   int 21h
   mov ah,01h
   int 21h
   cmp al,'N'
   jz no
   jmp cont

no:mov ah,4ch
   int 21h
```

```
Code Ends
End start
```

Output:

```
C:\>project.exe

Enter the series(terminated by Enter): 55 15 01 71 02 45 32 24

Ascending/Descending Order(A/D): A
Your ordered series is: 01 02 15 24 32 45 55 71
Do you want to continue?(Y/N): Y
Enter the series(terminated by Enter): 0B 0A 0F FE FD CE

Ascending/Descending Order(A/D): D
Your ordered series is: FE FD CE 0F 0B 0A
Do you want to continue?(Y/N): N
C:\>
```