AIDS MICROPROCESSOR LAB S21 BATCH (2023-24)

Experiment 4(a) Title: Assembly language programming to sort numbers in ascending order using software tool TASM 1.4

Name of student: Meet Raut   Class Roll Number: 2201084

Date of Performance: 04/03/2024

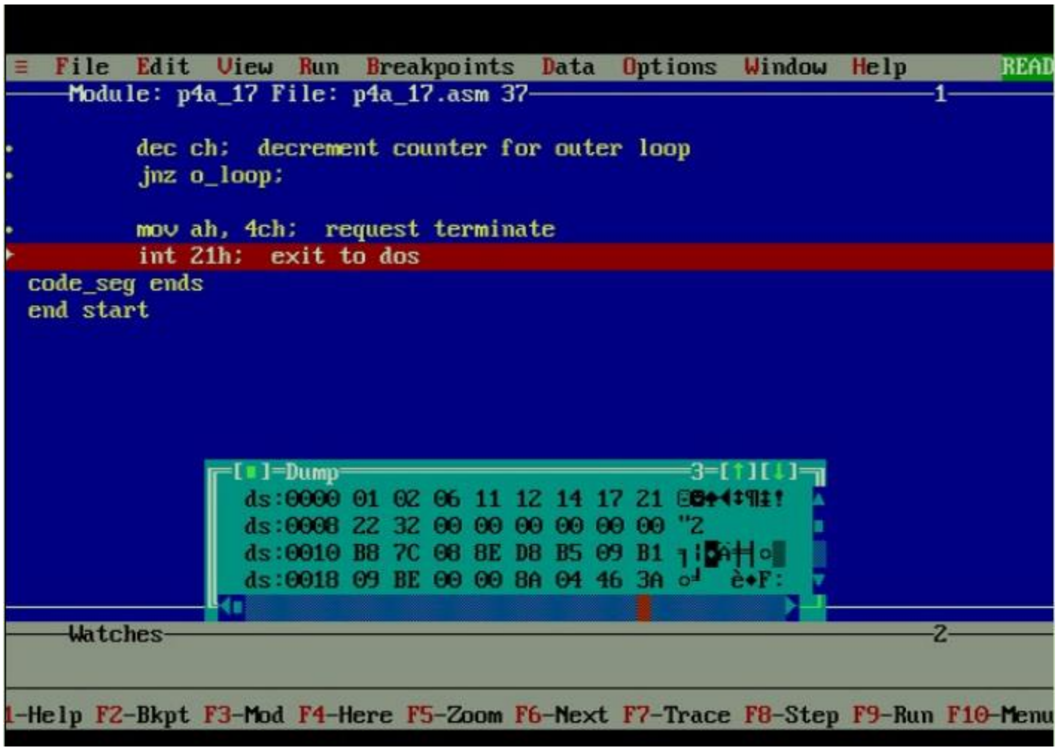
Batch: S2-1   Timing: 3:00-5:00   Date of Submission: 04/03/2024

Assembly language code

```
data_seg segment
    ary db 14h, 12h, 21h, 17h, 06h, 01h, 32h, 2h, 22h, 11h

data_seg ends

code_seg segment

assume cs:code_seg, ds:data_seg            #initialise segment
registers

start:

mov ax,data_seg                           #initialise data segment
register

 mov ds,ax;

 mov ch, 09h                          #initialise the counter for
outer loop

o_loop: mov cl, 09h                   #initialise the counter for the inner loop

            mov si, offset ary            #initialise SI to point to the first element of
            the array

i_loop: mov al, [si]           #fetch element pointed by SI for
            comparison inc si                        #update SI
```

```
         cmp al,[si]  #compare element pointed by SI to current
         maximum jbe swap       #if element is found <= current
         max, skip the swap xchg al,[si]      #swap if new
         maximum is found

         mov [si-1],al

swap:    dec cl                #decrement the counter for inner loop

         jnz i_loop;

         dec ch                    #decrement counter for
         outer loop jnz o_loop;

         mov ah, 4ch              #request
terminate

         int 21h                  #exit to dos

         code_seg ends

         end start
```

Result :


≡ File Edit View Run Breakpoints Data Options Window Help READ
Watches
1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
Module: p4a_17 File: p4a_17.asm 37                                    1

        dec ch;  decrement counter for outer loop
        jnz o_loop;

        mov ah, 4ch;  request terminate
        int 21h;  exit to dos
code_seg ends
end start
```

```
[■]=Dump                               3=[↑][↓]
ds:0000 01 02 06 11 12 14 17 21 ☺☻♠◄↕¶↨!
ds:0008 22 32 00 00 00 00 00 00 "2
ds:0010 B8 7C 08 8E D8 B5 09 B1 ╕|◘Ä╪╡○▒
ds:0018 09 BE 00 00 8A 04 46 3A ○╛  è♦F:
```

<u>Experiment 4(b) Title: Assembly language programming to sort numbers in descending order using software tool TASM 1.4</u>

Name of student: Meet Raut   Class Roll Number: 2201084

Date of Performance: 04/03/2024

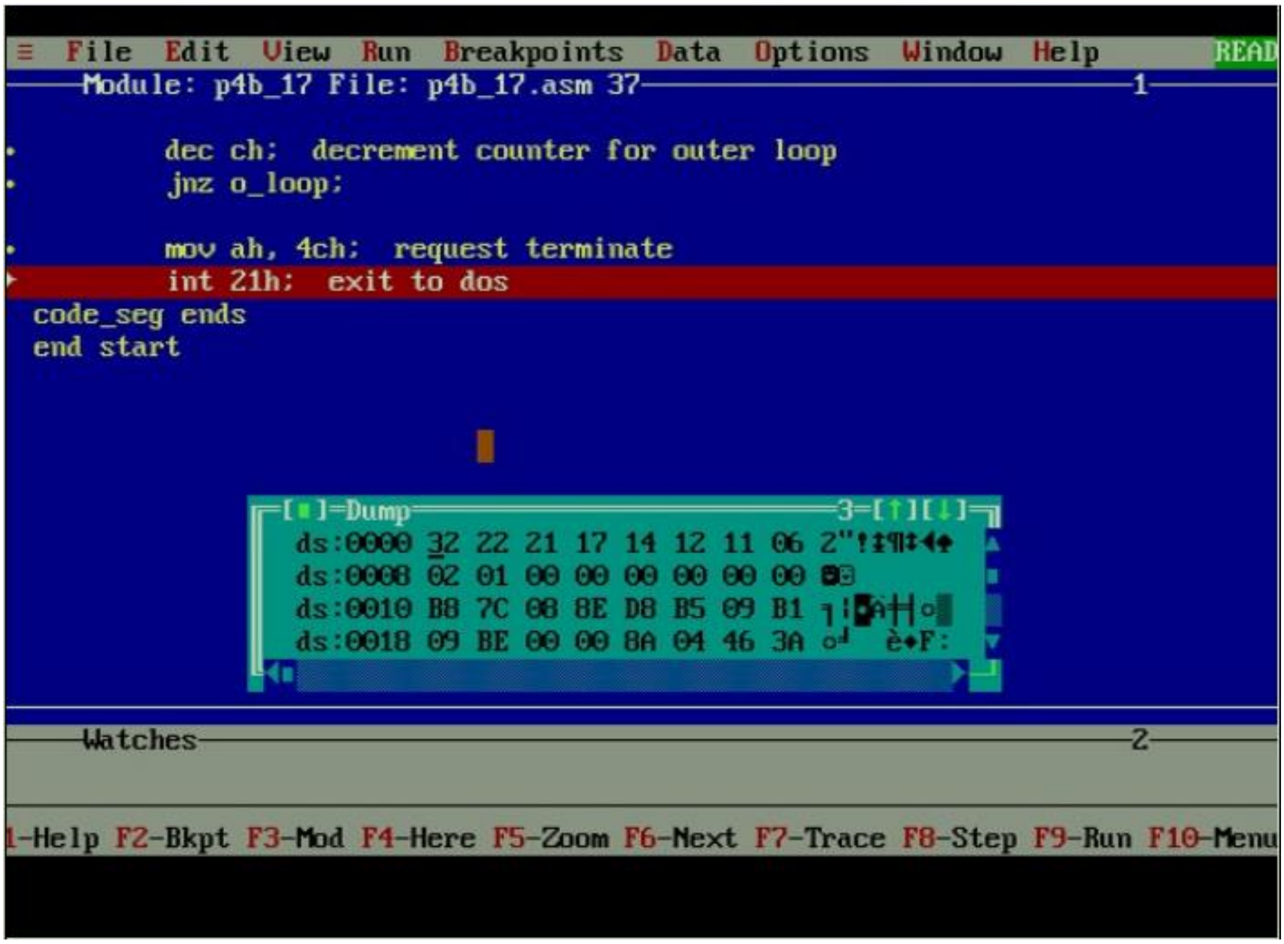
Batch: S2-1   Timing: 3:00-5:00   Date of Submission: 04/03/2024

<u>Assembly language code</u>

```
data_seg segment
   ary db 14h, 12h, 21h, 17h, 06h, 01h, 32h, 2h, 22h, 11h
data_seg ends
code_seg segment
assume cs:code_seg, ds:data_seg          #initialise segment registers
start:  mov ax,data_seg                  #initialise data segment register
```

```
mov ds,ax;
mov ch, 09h                     #initialise the counter for outer loop
o_loop: mov cl, 09h             #initialise the counter for the inner loop
mov si, offset ary             #initialise SI to point to the first element of the array
i_loop: mov al, [si]             #fetch element pointed by SI for comparison
inc si                          #update SI
cmp al,[si]                   #compare element pointed by SI to current maximum
jae swap                     #if element is found <= current max, skip the swap
xchg al,[si]                  #swap if new maximum is found
mov [si-1],al
swap:   dec cl                     #decrement the counter for inner loop
jnz i_loop
dec ch                             #decrement counter for outer loop
jnz o_loop
mov ah, 4ch                       #request terminate
int 21h                             #exit to dos
code_seg ends
end start
```

Result:



≡ File Edit View Run Breakpoints Data Options Window Help READ
Module: p4b_17 File: p4b_17.asm 37
1
Dump
3
Watches
2
1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
        dec ch;  decrement counter for outer loop
        jnz o_loop;

        mov ah, 4ch;  request terminate
        int 21h;  exit to dos
code_seg ends
end start
```

```
ds:0000 32 22 21 17 14 12 11 06 2"!‡¶↕◄♠
ds:0008 02 01 00 00 00 00 00 00 ☻☺
ds:0010 B8 7C 08 8E D8 B5 09 B1 ╕|◘Ä╪╡○▒
ds:0018 09 BE 00 00 8A 04 46 3A ○╛  è♦F:
```

CONCLUSION: LO 2, LO 3 mapped.

-------------------------------*****************************---------------------------------