



Academic Year: 2022-2023

Name:	Prerna Sunil Jadhav
Sap Id:	60004220127
Class:	T. Y. B.Tech (Computer Engineering)
Course:	Processor Organization and Architecture (POA)
Course Code:	DJ19CEL502
Experiment No.:	06

AIM: Assembly program to sort numbers in ascending and descending order.

CODE:

DESCENDING ORDER:

```
data segment
string1 db 99h,12h,56h,45h,36h data ends
code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ch,04h

up2:
mov cl,04h
lea si,string1

up1:
mov al,[si]
mov bl,[si+1]
cmp al,bl
jnc down
mov dl,[si+1]
xchg [si],dl
mov [si+1],dl

down:
inc si
dec cl
jnz up1
dec ch
jnz up2
int 3
code ends
end start
```

ASCENDING ORDER:

```
data segment
string1 db 99h,12h,56h,45h,36h data ends
code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov ch,04h

up2:
mov cl,04h
lea si,string1
up1:
mov al,[si]
mov bl,[si+1]
cmp al,bl

jc down
mov dl,[si+1]
xchg [si],dl
mov [si+1],dl

down:
inc si
dec cl

jnz up1
dec ch
jnz up2
int 3

code ends
end start
```



Academic Year: 2022-2023

OUTPUT:

The screenshot shows an x86 emulator window titled 'emulator: noname.exe'. The registers window on the left shows the following values:

Register	H	L
AX	07	36
BX	00	12
CX	00	00
DX	00	12
CS	F400	
IP	0104	
SS	0710	
SP	FFFA	
BP	0000	
SI	0004	
DI	0000	
DS	0710	
ES	0700	

The memory window shows the following data:

Address	Hex	ASCII
0710:0000	99 56 45 36 12 00 00 00-00 00 00 00 00 00 00 00 00	00E6t.....
0710:0010	B8 10 07 8E D8 B5 04 B1-04 BE 00 00 8A 04 8A 5C	7p.A+...e
0710:0020	01 3A C3 73 08 8A 54 01-86 14 88 54 01 46 FE C9	0:ts.eTCAWETOFI
0710:0030	75 EA FE CD 75 E1 CC 90-90 90 90 90 90 90 90	uN!u0jEEEEEEEE
0710:0040	90 90 90 90 90 90 90 90-90 90 90 F4 00 00 00 00	EEEEEEEEEEEE r...
0710:0050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

The assembly window shows the following code:

```
BIOS DI
INT 0FFh
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

The screenshot shows the same x86 emulator window, but with different register values and memory data.

The registers window shows the following values:

Register	H	L
AX	07	56
BX	00	99
CX	00	00
DX	00	45
CS	F400	
IP	0104	
SS	0710	
SP	FFFA	
BP	0000	
SI	0004	
DI	0000	
DS	0710	
ES	0700	

The memory window shows the following data:

Address	Hex	ASCII
0710:0000	12 36 45 56 99 00 00 00-00 00 00 00 00 00 00 00	\$EUV0.....
0710:0010	B8 10 07 8E D8 B5 04 B1-04 BE 00 00 8A 04 8A 5C	7p.A+...e
0710:0020	01 3A C3 72 08 8A 54 01-86 14 88 54 01 46 FE C9	0:ts.eTCAWETOFI
0710:0030	75 EA FE CD 75 E1 CC 90-90 90 90 90 90 90 90	uN!u0jEEEEEEEE
0710:0040	90 90 90 90 90 90 90 90-90 90 90 F4 00 00 00 00	EEEEEEEEEEEE r...
0710:0050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0710:0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

The assembly window shows the following code:

```
01 DATA SEGMENT
02 STRING1 DB 99H,12H,56H,4
03 DATA ENDS
04 CODE SEGMENT
05 ASSUME CS:CODE,DS:DATA
06 START: MOV AX,DATA
07 MOV DS,AX
08 MOV CH,04H
09 UP2: MOV CL,04H
10 LEA SI,STRING1
11
12
13 UP1:MOV AL,[SI]
14 MOV BL,[SI+1]
15 CMP AL,BL
16 JC DOWN
17 MOV DL,[SI+1]
18 XCHG [SI],DL
19 MOV [SI+1],DL
20
21 DOWN: INC SI
22 DEC CL
23 JNZ UP1
24 DEC CH
25 JNZ UP2
26 INT 3
27 CODE ENDS
28 END START
```