MICROPROCESSOR ASSIGNMENTS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIVISION : B

ROLL NO 222006

PROBLEM STATEMENT :

Write X86 program to sort the list of integers in ascending/descending order. Read the input from the text file and write the sorted data back to the same text file using bubble sort.

CODE :

section .data

title: db 0x0A,"\*\*\*\*\* BuBbLe SoRt \*\*\*\*\*", 10

title\_len: equ $-title

openmsg: db "File Opened Successfully",0x0A

openmsg\_len: equ $-openmsg

closemsg: db "File Closed Successfully",0x0A

closemsg\_len: equ $-closemsg

errormsg: db "Failed to open file", 0x0A

errormsg\_len: equ $-errormsg

sortmsg: db "After Sorting "

sortmsg\_len: equ $-sortmsg

f1name: db 'file.txt', 0

%macro scall 4

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

section .bss

buffer: resb 200

bufercpy: resb 200

bufferlen:resb 8

cnt1:resb 8

cnt2: resb 8

fdis:resb 8

Section .text

global \_start

\_start:

scall 1,1,title,title\_len

scall 2,f1name,2,777 ;Opening file

mov qword[fdis],rax ;RAX contains file descriptor value

cmp rax,-1H

jle ERROR

scall 1,1,openmsg,openmsg\_len

jmp next1

ERROR:

scall 1,1,errormsg,errormsg\_len

jmp EXIT

next1:

scall 0,[fdis],buffer,200 ;reading contents of file in buffer

;rax contains actual number of bytes read

mov qword[bufferlen],rax ;for rounds

mov qword[cnt1],rax

mov qword[cnt2],rax

BUBBLE:

mov al,byte[cnt2]

mov byte[cnt1],al

dec byte[cnt1]

mov rsi,buffer

mov rdi,buffer+1

loop:

mov bl,byte[rsi]

mov cl,byte[rdi]

cmp bl,cl

ja SWAP

inc rsi

inc rdi

dec byte[cnt1]

jnz loop

dec byte[bufferlen]

jnz BUBBLE

jmp END

SWAP:

mov byte[rsi],cl

mov byte[rdi],bl

inc rsi

inc rdi

dec byte[cnt1]

jnz loop

dec byte[bufferlen]

jnz BUBBLE

END:

scall 1,1,sortmsg,sortmsg\_len

scall 1,1, buffer,qword[cnt2]

scall 1,qword[fdis],sortmsg,sortmsg\_len ;writing to file.txt

scall 1,qword[fdis],buffer,qword[cnt2] ;writing to file2.txt

;Closing file2

mov rax,3

mov rdi,f1name

scall 1,1,closemsg,closemsg\_len

EXIT:

mov rax,60

mov rdi,0

syscall

OUTPUT :

