**Assignment number: 8**

**Subject: MICROPROCESSOR LAB**

Name: ***RIA MITTAL***

Class: ***SECOND YEAR ENGINEERING***

Division: ***B***

Roll no: ***222008***

Batch: ***B1***

**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:**

%include "macro.asm"

;------------------------------------------------------------------------

section .data

nline db 10

nline\_len equ $-nline

ano db 10,10,10,10,"Bubble sort using file operations"

db 10,"---------------------------------------------------",10

ano\_len equ $-ano

filemsg db 10,"Enter filename of input data : "

filemsg\_len equ $-filemsg

omsg db 10,"Sorting using bubble sort Operation successful."

db 10,"Output stored in same file...",10,10

omsg\_len equ $-omsg

errmsg db 10,"ERROR in opening/reading/writing File...",10

errmsg\_len equ $-errmsg

ermsg db 10,"ERROR in writing File...",10

ermsg\_len equ $-ermsg

exitmsg db 10,10,"Exit from program...",10,10

exitmsg\_len equ $-exitmsg

;---------------------------------------------------------------------------

section .bss

buf resb 1024

buf\_len equ $-buf ; buffer length

filename resb 50

filehandle resq 1

abuf\_len resq 1 ; actual buffer length

array resb 10

n resq 1

;--------------------------------------------------------------------------

section .text

global \_start

\_start:

print ano,ano\_len ;assignment no.

print filemsg,filemsg\_len

read filename,50

dec rax

mov byte[filename + rax],0 ; blank char/null char

fopen filename ; on succes returns handle

cmp rax,-1H ; on failure returns -1

je Error

mov [filehandle],rax

fread [filehandle],buf, buf\_len

dec rax ; EOF

mov [abuf\_len],rax

call bsort

jmp Exit

Error: print errmsg, errmsg\_len

Exit: print exitmsg,exitmsg\_len

exit

;--------------------------------------------------------------------------------

bsort: ; Bubble sort procedure

call buf\_array

xor rax,rax

mov rbp,[n]

dec rbp

xor rcx,rcx

xor rdx,rdx

xor rsi,rsi

xor rdi,rdi

mov rcx,0 ; i=0

oloop: mov rbx,0 ; j=0

mov rsi,array ; a[j]

iloop: mov rdi,rsi ; a[j+1]

inc rdi

mov al,[rsi]

cmp al,[rdi]

jbe next

mov dl,0

mov dl,[rdi] ; swap

mov [rdi],al

mov [rsi],dl

next: inc rsi

inc rbx ; j++

cmp rbx,rbp

jb iloop

inc rcx

cmp rcx,rbp

jb oloop

fwrite [filehandle],omsg, omsg\_len

fwrite [filehandle],array,[n]

fclose [filehandle]

print omsg, omsg\_len

print array,[n]

RET

Error1:

print ermsg, ermsg\_len

RET

;------------------------------------------------------------------

buf\_array:

xor rcx,rcx

xor rsi,rsi

xor rdi,rdi

mov rcx,[abuf\_len]

mov rsi,buf

mov rdi,array

next\_num:

mov al,[rsi]

mov [rdi],al

inc rsi ; number

inc rsi ; newline

inc rdi

inc byte[n] ; counter

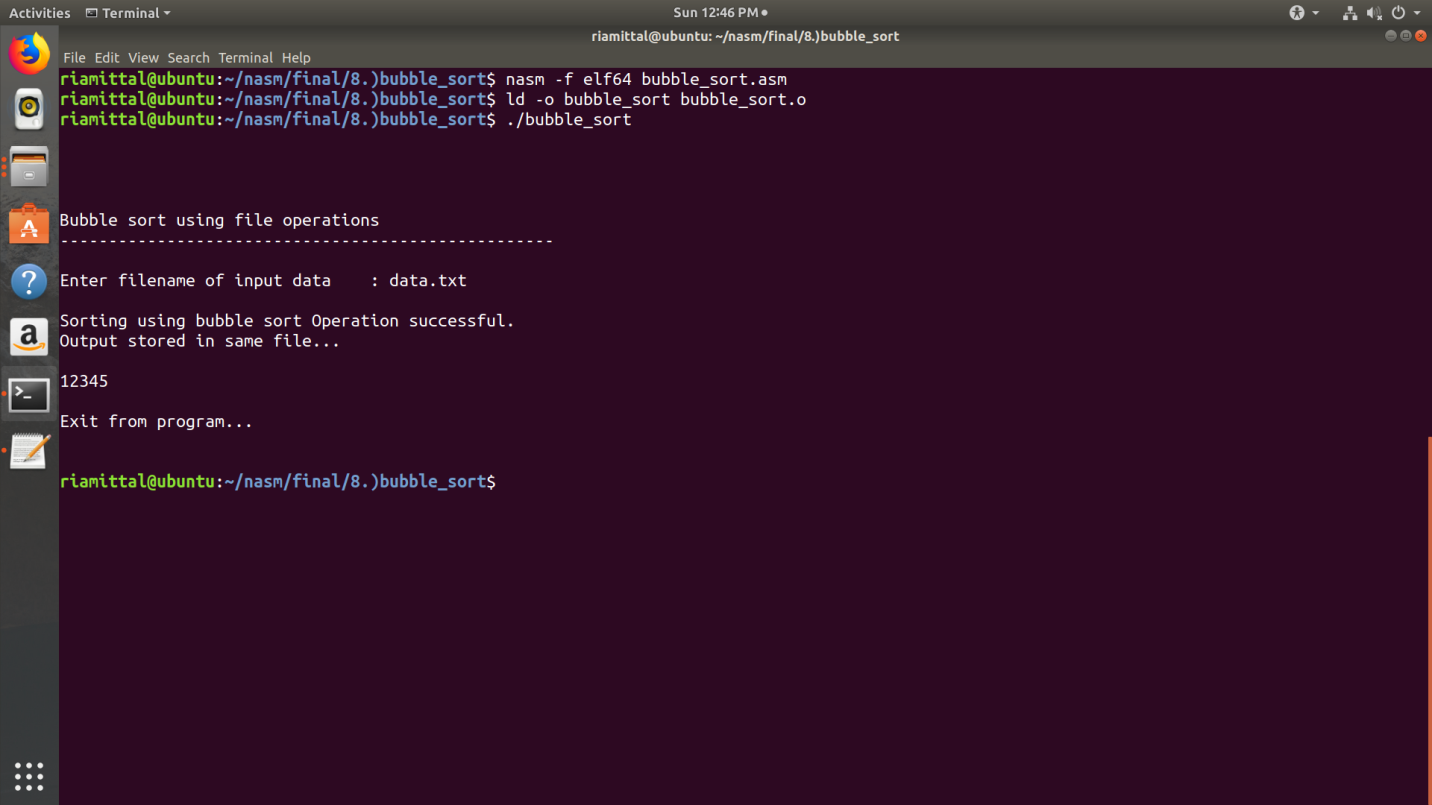
dec rcx ; number

dec rcx ; newline

jnz next\_num

ret

;------------------------------------------------------------------

****