;Write X86 ALP to find, a) Number of Blank spaces

; b) Number of lines

; c) Occurrence of a particular character.

; Accept the data from the text file.

;The text file has to be accessed **during Program_1 execution ;and write FAR PROCEDURES**

; in **Program_2 f**or the rest of the processing.

;Use of **PUBLIC and EXTERN** directives is mandatory.

;********** **p1.asm file** **********

```
section .data
global msg6,len6,scount,ncount,occcount,new,new_len
fname: db 'abc.txt',0
msg: db "File opened successfully",0x0A
len: equ $-msg
msg1: db "File closed successfully",0x0A
len1: equ $-msg1
msg2: db "Error in opening file",0x0A
len2: equ $-msg2
msg3: db "No of Blank spaces :",0x0A
len3: equ $-msg3
msg4: db "No of NewLines:",0x0A
len4: equ $-msg4
msg5: db "Enter character:",0x0A
len5: equ $-msg5
```

```
msg6: db "No of occurrences character:",0x0A
len6: equ $-msg6
new: db "",0x0A
new_len: equ $-new

scount: db 0            ;spaces
ncount: db 0            ;NewLines
ccount: db 0
chacount: db 0          ;char

section .bss
global cnt,cnt2,cnt3,buffer
; global variables created in .data and .bss sections but  declared outside the same segment
fd resb 17                          ;file descriptor
buffer  resb 200
buf_len  resb 17
cnt  resb 2
cnt2  resb 2
cnt3 resb 2
occr resb 2

%macro scall 4          ; macro call for RW
mov rax,%1
mov rdi,%2
```

```
mov rsi,%3
mov rdx,%4
syscall
%endmacro

section .text
global _start
_start:

extern spaces, enters, occ          ; extern directive identifies Proc/variable defined in another source module
mov rax,2                   ;open file cursor goes in end of file
mov rdi, fname                  ;file name as second parameter
mov rsi,2                   ;0=read only,1=write only 2=read/write mode
mov rdx,0777                ; Setting permission for read, write and execute by all
syscall

mov qword[fd],rax
BT rax,63                   ; CF=0  read file
jc next                     ;CF=1
scall 1,1,msg,len           ;File open successfully
jmp next2
next: scall 1,1,msg2,len2               ;CF=1 or Error to open file
jmp exit
```

```
next2:scall 0,[fd],buffer, 200      ;macro call to read from file
mov qword[buf_len],rax
mov qword[cnt],rax
mov qword[cnt2],rax
mov qword[cnt3],rax

scall 1,1,msg3,len3                 ;No of Blank spaces :
call spaces
scall 1,1,msg4,len4                 ;No of NewLines:
call enters
scall 1,1,msg5,len5                 ;Enter character:
scall 0,1,occr,2                    ; read and print  input chr
mov bl, byte[occr]
call occ


   scall 1,1,msg1,len1              ;file close successfully
   mov     rax, 3                   ;close Fname (abc.txt)
   mov     rdi, fname
   syscall

 exit:mov rax,60                    ;Program end
   mov rdi,0
   syscall
```

```
;********* **********P2 file ****************
;P2.asm
section .data
extern msg6,len6,scount,ncount,occrance,new,new_len
section .bss
extern cnt,cnt2,cnt3,scall,buffer

%macro scall 4
mov rax,%1
mov rdi,%2
mov rsi,%3
mov rdx,%4
syscall
%endmacro

section .text
global main2
main2:
global spaces,enters,occ     ; globally get called FAR_PROC for spaces,enters,occ

;************checking number of spaces *************
spaces:mov rsi,buffer
up:mov al, byte[rsi]
cmp al,20H    ; space character (ASCII code 20h)
```

```
je next3
inc rsi
dec byte[cnt]
jnz up
jmp next4
next3:inc rsi
inc byte[scount]     ;increment space count
dec byte[cnt]
jnz up

next4:add byte[scount], 30h       ; hex to ASCII
scall 1,1,scount, 2      ; result of no of spaces count
scall 1,1,new,new_len
ret


; ************ check new line ****************
enters:mov rsi,buffer
up2:
mov al, byte[rsi]
cmp al,0AH          ;check enter key = 0A or 10 (linefeed or /n)
je next5
inc rsi
dec byte[cnt2]
jnz up2
```

```
jmp next6
next5:inc rsi
inc byte[ncount]            ;new line counter increment
dec byte[cnt2]
jnz up2
next6:add byte[ncount], 30h        ; hex to ASCII
scall 1,1,ncount, 2                ; result of new line count
scall 1,1,new,new_len
ret

;********** occurrence of character *****************
occ:mov rsi,buffer
up3:mov al, byte[rsi]
cmp al,bl                  ; bl = read input chr and al=no of characters in file buffer ;cmp both
je next7
inc rsi
dec byte[cnt3]
jnz up3
jmp next8
next7:inc rsi
inc byte[occrance]
dec byte[cnt3]
jnz up3
next8:add byte[occrance], 30h          ; hex to ASCII
```

```
scall 1,1,msg6,len6      ;print No. of char occurrence msg
scall 1,1,occrance, 1    ; result of No. of char occurrence
scall 1,1,new,new_len
ret
;***** ************p2.asm file end ****************

;***********Text file (abc.txt)************
;Hello
;Welcome to Pune
;This is microprocessor Lab

;*******output*******
; nasm -f elf64 p1 p1.asm
; nasm -f elf64 p2 p2.asm
; ld -o p p1.o p2.o
; ./p
;File opened successfully
;Spaces:6
;NewLines:3
;Enter character:e
;No of occurances:5
;file closed successfully
```