;nasm 2.13.02

section .data

nLine db 0xA, 0xD

nLineLen equ $-nLine

section .bss

var: resb 1

var1: resb 1

var2: resb 1

section .text

global \_start

\_start:

;initializing variables

mov [var1],byte 0

mov [var2],byte 0

read:

;reading into var

mov eax,3

mov ebx,0

mov ecx,var

mov edx,1

int 80h

cmp [var],byte '.' ;comparing var to '.'

je end ;if var is '.', end program

cmp [var],byte 65 ;comparing var to 'A'

jl numb ;if var < 65, we have a number

jg Lrge ;if var > 65, we have a letter

Lrge:

add [var1],byte 1 ;counting letters read

jmp read ;jumping back to read next byte

numb:

add [var2],byte 1 ;counting numbers read

jmp read ;jumping back to read next byte

end:

;adding ascii 0 so when printed will print correct character

add [var1],byte '0'

add [var2],byte '0'

;printing the letter count

mov eax,4

mov ebx,1

mov ecx,var1

mov edx,1

int 80h

;printing a new line

mov eax,4

mov ebx,1

mov ecx,nLine

mov edx,nLineLen

int 80h

;printing the number count

mov eax,4

mov ebx,1

mov ecx,var2

mov edx,1

int 80h

;ending the program

mov eax,1

mov ebx,0

int 80h